Advances in Applied Mathematics **25,** 65–101 (2000)
doi:10.1006/aama.2000.0688, available online at http://www.idealibrary.com on IDEAL®

# Perfect Two-Fault Tolerant Search with Minimum Adaptiveness[1]

Ferdinando Cicalese[2]



E-mail: cicalese@dia.unisa.it

and

Daniele Mundici[3]

*Department of Computer Science, University of Milan,*
*Via Comelico 39-41, 20135 Milan, Italy*
E-mail: mundici@mailserver.unimi.it

Received September 11, 1999; accepted March 8, 2000

Our aim is to minimize the number of answer/question alternations in a perfect two-fault-tolerant search. Ulam and Rényi posed the problem of searching for an unknown $m$-bit number by asking the minimum number of *yes–no* questions, when up to $\ell$ of the answers may be erroneous/mendacious. Berlekamp considered the same problem in the context of error-correcting communication with feedback. Among others, he proved that at least $q_\ell(m)$ questions are necessary, where $q_\ell(m)$ is the smallest integer $q$ satisfying $2^{q-m} \geq \sum_{j=0}^{\ell} \binom{q}{j}$. When all questions are asked in advance, and adaptiveness has no role, finding a *perfect* strategy (i.e., a strategy with $q_\ell(m)$ questions) amounts to finding an $\ell$-error-correcting code with $2^m$ codewords of length $q_\ell(m)$. From coding theory it is known that such perfect *non-adaptive* searching strategies are rather the exception, for $\ell \geq 2$. At the other extreme, in a *fully adaptive* search, where the $(t+1)$th question is asked knowing the answer to the $t$th question, perfect strategies are known to exist for all sufficiently large $m$.

[1] A preliminary draft of the first part of this paper, only dealing with binary search, appears in [7].

[2] Partially supported by an ENEA grant.

[3] Partially supported by COST ACTION 15 on many-valued logics for computer science applications, and by the Italian MURST Project on Logic.



0196-8858/00 $35.00

What happens if we impose restrictions on the amount of adaptiveness avaliable to the questioner? Focusing attention on the case $\ell = 2$, we shall prove that, for each $m \neq 2$, perfect searching strategies still exist even if the questioner is allowed to adapt his strategy *only once*. All our results are constructive and *explicitly* yield perfect two-error-correcting codes with the least possible feedback. We finally generalize our results to $k$-ary search. 

*Key Words:* searching with errors; fault-tolerant search; adaptive search; perfect coding; error-correcting codes; communication with feedback.

## 1. INTRODUCTION

Let $q_\ell(m)$ be the smallest integer $q$ satisfying $2^{q-m} \geq \sum_{j=0}^{\ell} \binom{q}{j}$. The first aim of this paper is to *explicitly give, for each* $m \neq 2$, perfect two-error-correcting search strategies over the space of $m$-bit numbers, with the least possible degree of adaptiveness/feedback. The second aim is to generalize these results to $k$-ary search.

The problem of efficient search of an unknown element in a finite set $S$ is often reformulated as a game between two players—one deciding the questions to be asked, and the other deciding the answering strategy that makes as hard as possible the first player's task. In Berlekamp's theory of error-correcting communication with feedback [3] (also see [11, 19]) one further assumes answers to be subject to distortion. Variants of the 20 questions game with lies yield the game-theoretic counterpart of the corresponding search problem.

We shall be concerned with the following problem: Two players, called Paul and Carole, first fix a set $S = \{0, 1, \ldots, 2^m - 1\}$. Now Carole thinks of a number $x_{\text{Carole}} \in S$, and Paul must find out $x_{\text{Carole}}$ by asking questions, to which Carole can only answer "yes" or "no." Assuming Carole is allowed to lie—or just to be inaccurate—in up to $\ell$ answers, what is the minimum number of questions needed by Paul to infallibly guess $x_{\text{Carole}}$?

When the questions are asked *adaptively*, i.e., the $i$th question is asked knowing the answer to the $(i-1)$th question, the problem is generally referred to as the Ulam–Rényi problem, [18, p. 47; 23, p. 281]. Optimal solutions (for each $m$) are given in [9, 15, 16, 21], respectively, for the cases $\ell = 1$, $\ell = 2$, $\ell = 3$, and (for all sufficiently large $m$) for the general case. See [10] for a survey. If queries with $k$ many possible answers are considered, one gets a $k$-ary search with lies. Solutions of the corresponding generalized Ulam–Rényi problems can be found in [1] for the case $\ell = 1$ and in [6, 8] for the case $\ell = 2$.

At the other, fully *non-adaptive* extreme, when all questions must be asked in advance, the Ulam–Rényi problem amounts to finding an $\ell$-error correcting code with $|S|$ codewords of shortest length, where $|S|$ denotes the number of elements of $S$. As is well known for $\ell = 1$ Hamming codes

yield searching strategies with the smallest possible number of questions—indeed, Pelc [17] shows that adaptiveness is irrelevant even under the additional assumption that repetition of the same question is forbidden. By contrast, for $\ell > 1$ the best known non-adaptive search strategies over the set of $m$-bit numbers generally require a number of questions strictly greater than $q_\ell(m)$ (see, e.g., [13, 22]).

In many practical applications where adaptiveness takes its toll, preference is given to search procedures involving large batches of non-adaptive questions. One can thus minimize the number of interactive alternations between answers and questions. For instance, in certain applications of computational molecular biology (see [12]) preference is given to two-stage searching strategies, where the search is adapted only once.

In this paper we give a detailed account of two-stage *perfect* two-fault tolerant searching strategies as follows: Paul first asks about the $m$ bits of $x_{\text{Carole}}$ and then, only depending on Carole's answers, he asks a second non-adaptive batch $\mathcal{Q}$ of $q_2(m) - m$ questions. A careful choice of $\mathcal{Q}$ allows Paul to infallibly guess $x_{\text{Carole}}$, even if up to two of Carole's $q_2(m)$ answers are false. We describe an inductive algorithm to effectively compute $\mathcal{Q}$ for all $m \neq 2$: this includes all cases of practical interest, as opposed to asymptotic results. To this purpose we extensively build on error-correcting codes existing in the literature (notably [4, 5]).

A substantial portion of this paper is devoted to extending these results to a $k$-ary search.

## 2. THE ULAM–RÉNYI GAME

*Questions*, *Answers*, *States*, *Strategies*

Assuming Carole and Paul to have agreed on the *search space* $S = \{0, 1, \ldots, 2^m - 1\}$, by a *question* $T$ we understand an arbitrary subset $T$ of $S$. The *opposite question* is the complement $S \setminus T$. In case Carole's answer is "yes," numbers in $T$ are said to *satisfy* Carole's answer, while numbers in $S \setminus T$ *falsify* it. Carole's negative answer to $T$ has the same effect as a positive answer to the opposite question $S \setminus T$.

Suppose questions $T_1, \ldots, T_t$ have been asked and answers $b_1, \ldots, b_t$ have been received from Carole ($b_i \in \{\text{no}, \text{yes}\}$). Since up to *two* of Carole's answers may be erroneous, a number $y \in S$ must be rejected from consideration if, and only if, it falsifies three or more answers. The remaining numbers of $S$ still are possible candidates for the unknown $x_{\text{Carole}}$. All that Paul knows (Paul's *state* of knowledge) is a triplet $\sigma = (A_0, A_1, A_2)$ of pairwise disjoint subsets of $S$, where $A_i$ is the set of numbers falsifying

$i$ answers, $i = 0, 1, 2$. The *initial* state is naturally given by $(S, \emptyset, \emptyset)$. A state $(A_0, A_1, A_2)$ is *final* iff $A_0 \cup A_1 \cup A_2$ is empty or has exactly one element.

For any state $\sigma = (A_0, A_1, A_2)$ and question $T \subseteq S$, the two states $\sigma^{\text{yes}}$ and $\sigma^{\text{no}}$, respectively, resulting from Carole's positive or negative answer, are given by

$$\sigma^{\text{yes}} = (A_0 \cap T,\ (A_0 \setminus T) \cup (A_1 \cap T),\ (A_1 \setminus T) \cup (A_2 \cap T)) \qquad (1)$$

and

$$\sigma^{\text{no}} = (A_0 \setminus T,\ (A_0 \cap T) \cup (A_1 \setminus T),\ (A_1 \cap T) \cup (A_2 \setminus T)). \qquad (2)$$

Turning attention to questions $T_1, \ldots, T_t$ and their respective answers $\vec{b} = b_1, \ldots, b_t$, iterated application of the above formulas yields a sequence of states

$$\sigma_0 = \sigma, \qquad \sigma_1 = \sigma_0^{b_1}, \qquad \sigma_2 = \sigma_1^{b_2}, \qquad \ldots, \qquad \sigma_t = \sigma_{t-1}^{b_t}. \qquad (3)$$

By a *strategy* $\mathcal{S}$ *with* $q$ *questions* we mean the full binary tree of depth $q$, where each node $\nu$ is mapped into a question $T_\nu$, and the two edges $\eta_{\text{left}}, \eta_{\text{right}}$ generated by $\nu$ are respectively labelled yes and no. Let $\vec{\eta} = \eta_1, \ldots, \eta_q$ be a path in $\mathcal{S}$, from the root to a leaf, with respective labels $b_1, \ldots, b_q$, generating nodes $\nu_1, \ldots, \nu_q$ and associated questions $T_{\nu_1}, \ldots, T_{\nu_q}$. Fix an arbitrary state $\sigma$. Then, iterated application of (1), (2) naturally transforms $\sigma$ into $\sigma^{\vec{\eta}}$ (where the dependence on the $b_j$ and $T_j$ is understood). We say that strategy $\mathcal{S}$ is *winning* for $\sigma$ iff for every path $\vec{\eta}$ the state $\sigma^{\vec{\eta}}$ is final.

A strategy is said to be *non-adaptive* iff all nodes at the same depth of the tree are mapped into the same question.

*Type*, *Weight*, *Character*, *Berlekamp's Lower Bound*

Let $\sigma = (A_0, A_1, A_2)$ be a state. For each $i = 0, 1, 2$ let $a_i = |A_i|$ be the number of elements of $A_i$. Then the triplet $(a_0, a_1, a_2)$ is called the *type* of $\sigma$. By definition, the *Berlekamp weight of* $\sigma$ *before* $q$ *questions*, $q = 0, 1, 2, \ldots$, is given by

$$w_q(\sigma) = a_0\left(\binom{q}{2} + q + 1\right) + a_1(q+1) + a_2. \qquad (4)$$

The *character* $\text{ch}(\sigma)$ of a state $\sigma$ is the smallest integer $q \geq 0$ such that $w_q(\sigma) \leq 2^q$.

By abuse of notation, the weight of *any* state $\sigma$ of type $(a_0, a_1, a_2)$ before $q$ questions will be denoted $w_q(a_0, a_1, a_2)$. Similarly, its character will also be denoted $\text{ch}(a_0, a_1, a_2)$.

As an immediate consequence we have the following monotonicity properties: For any two states $\sigma' = (A'_0, A'_1, A'_2)$ and $\sigma'' = (A''_0, A''_1, A''_2)$, respectively, of type $(a'_0, a'_1, a'_2)$ and $(a''_0, a''_1, a''_2)$, if $a'_i \le a''_i$ for all $i = 1, 2, 3$ then

$$\mathrm{ch}(\sigma') \le \mathrm{ch}(\sigma'') \qquad \text{and} \qquad w_q(\sigma') \le w_q(\sigma'') \tag{5}$$

for each $q \ge 0$. Note that $\mathrm{ch}(\sigma) = 0$ iff $\sigma$ is a final state. The proof of the following results goes back to [3]:

LEMMA 2.1. *Let $\sigma$ be an arbitrary state, and let $T \subseteq S$ be a question. Let $\sigma^{\mathrm{yes}}$ and $\sigma^{\mathrm{no}}$ be as in* (1), (2). *We then have*

(i) Conservation Law: *For every integer $q \ge 1$,*

$$w_q(\sigma) = w_{q-1}(\sigma^{\mathrm{yes}}) + w_{q-1}(\sigma^{\mathrm{no}}).$$

(ii) Berlekamp's lower bound: *If $\sigma$ has a winning strategy with $q$ questions then $q \ge \mathrm{ch}(\sigma)$.*

DEFINITION 2.2. A strategy $\mathcal{S}$ with $q$ questions for a state $\sigma$ is said to be *perfect* iff $\mathcal{S}$ is winning for $\sigma$ and $q = \mathrm{ch}(\sigma)$.[4]

Let $\sigma = (A_0, A_1, A_2)$ be a state. Let $T \subseteq S$ be a question. We say that $T$ is *balanced for* $\sigma$ iff $|A_j \cap T| = |A_j \setminus T|$, for each $j = 0, 1, 2$.

LEMMA 2.3. *Let $T$ be a balanced question for a state $\sigma = (A_0, A_1, A_2)$. Let $n = \mathrm{ch}(\sigma)$. Let $\sigma^{\mathrm{yes}}$ and $\sigma^{\mathrm{no}}$ be as in* (1), (2) *above. Then, for each integer $q \ge 0$,*

(i) $w_q(\sigma^{\mathrm{yes}}) = w_q(\sigma^{\mathrm{no}})$,

(ii) $\mathrm{ch}(\sigma^{\mathrm{yes}}) = \mathrm{ch}(\sigma^{\mathrm{no}}) = n - 1$.

*Proof.* Condition (i) is an immediate consequence of the definition of Berlekamp's weight, together with (1), (2). In order to prove (ii), since for each $q$, $w_q(\sigma^{\mathrm{yes}}) = w_q(\sigma^{\mathrm{no}})$, then by Lemma 2.1(i) we have $2^n \ge w_n(\sigma) = w_{n-1}(\sigma^{\mathrm{yes}}) + w_{n-1}(\sigma^{\mathrm{no}}) = 2w_{n-1}(\sigma^{\mathrm{yes}}) = 2w_{n-1}(\sigma^{\mathrm{no}})$ and $2^{n-1} < w_{n-1}(\sigma) = w_{n-2}(\sigma^{\mathrm{yes}}) + w_{n-2}(\sigma^{\mathrm{no}}) = 2w_{n-2}(\sigma^{\mathrm{yes}}) = 2w_{n-2}(\sigma^{\mathrm{no}})$, whence $w_{n-1}(\sigma^{\mathrm{yes}}) = w_{n-1}(\sigma^{\mathrm{no}}) \le 2^{n-1}$ and $w_{n-2}(\sigma^{\mathrm{yes}}) = w_{n-2}(\sigma^{\mathrm{no}}) > 2^{n-2}$; i.e., $\mathrm{ch}(\sigma^{\mathrm{yes}}) = \mathrm{ch}(\sigma^{\mathrm{no}}) = n - 1$. ∎

[4]Because a perfect strategy $\mathcal{S}$ uses the least possible number of questions, as given by Berlekamp's bound, $\mathcal{S}$ cannot be superseded by a shorter strategy. Thus every *perfect* strategy is a fortiori an *optimal* strategy. On the other hand, this paper will exhibit several optimal strategies which are not perfect.

## 3. BACKGROUND FROM CODING THEORY

For arbitrary integers $k \geq 2$ and $n > 0$ let $\vec{x}, \vec{y} \in \{0, 1, \ldots, k-1\}^n$. The *Hamming distance* $d_H(\vec{x}, \vec{y})$ is defined by

$$d_H(\vec{x}, \vec{y}) = |\{i \in \{1, \ldots, n\} \mid x_i \neq y_i\}|,$$

where, as above, $|A|$ denotes the number of elements of $A$.

The *Hamming sphere* $\mathscr{B}_r(\vec{x})$ *with radius r and center* $\vec{x}$ is the set of elements of $\{0, 1, \ldots, k-1\}^n$ whose Hamming distance from $\vec{x}$ is $\leq r$; in symbols,

$$\mathscr{B}_r(\vec{x}) = \{\vec{y} \in \{0, 1, \ldots, k-1\}^n \mid d_H(\vec{x}, \vec{y}) \leq r\}.$$

For any $\vec{x} \in \{0, 1, \ldots, k-1\}^n$ we have

$$|\mathscr{B}_r(\vec{x})| = \sum_{i=0}^{r} \binom{n}{i} (k-1)^i. \tag{6}$$

The *Hamming weight* $w_H(\vec{x})$ is the number of non-zero digits of $\vec{x}$.

We refer to [13] for background in coding theory. When $k$ is clearly understood from the context, by a code we shall mean a $k$-ary code in the following sense:

DEFINITION 3.1. A *(k-ary) code* $\mathscr{C}$ *of length n* is a subset of $\{0, 1, \ldots, k-1\}^n$. When $k = 2$ we will call $\mathscr{C}$ a *binary* code. Its elements are called *codewords*. The *minimum distance* of $\mathscr{C}$ is given by

$$\delta(\mathscr{C}) = \min\{d_H(\vec{x}, \vec{y}) \mid \vec{x}, \vec{y} \in \mathscr{C}, \vec{x} \neq \vec{y}\}.$$

We say that $\mathscr{C}$ is an $(n, m, d)$ *code* iff $\mathscr{C}$ has length $n$, $|\mathscr{C}| = m$, and $\delta(\mathscr{C}) = d$. The *minimum weight* of $\mathscr{C}$ is the minimum of the Hamming weights of its codewords; in symbols, $\mu(\mathscr{C}) = \min\{w_H(\vec{x}) \mid \vec{x} \in \mathscr{C}\}$.

Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be two codes of length $n$. The *minimum distance between* $\mathscr{C}_1$ *and* $\mathscr{C}_2$ is defined by $\Delta(\mathscr{C}_1, \mathscr{C}_2) = \min\{d_H(\vec{x}, \vec{y}) \mid \vec{x} \in \mathscr{C}_1, \vec{y} \in \mathscr{C}_2\}$.

By definition, the empty set $\varnothing$ is an $(n, 0, d)$ $k$-ary code for all integers $n, d \geq 0$ and $k \geq 2$. Further, for any code $\mathscr{C}$ and integer $d \geq 0$, we have the inequality $\Delta(\varnothing, \mathscr{C}) \geq d$. Similarly, the code consisting of the single codeword $\underbrace{0 \cdots 0}_{n \text{ times}}$ is an $(n, 1, d)$ $k$-ary code for all integers $d \geq 0$ and $k \geq 2$.

LEMMA 3.2. *Let* $e, n, m$ *be integers* $> 0$, *and let* $k \geq 2$. *Suppose* $\mathscr{C}$ *to be an* $(n, m, d)$ $k$*-ary code such that* $\mu(\mathscr{C}) \geq e$ *and* $d \geq 3$. *Then there exists an* $(n+2, km, d')$ $k$*-ary code* $\mathscr{D}$ *such that* $\mu(\mathscr{D}) \geq e$ *and* $d' \geq 3$.

*Proof.* Given any code $\mathcal{G}$ of length $n$ together with tuples $\vec{x} = x_1 \cdots x_n \in \{0, 1, \ldots, k-1\}^n$ and $\vec{a} = a_1 a_2 \cdots a_s \in \{0, 1, \ldots, k-1\}^s$, we denote by $\{\mathcal{G} \oplus \vec{x}\} \otimes \vec{a}$ the $k$-ary code of length $n+s$ whose codewords are obtained by adding $\vec{x}$ (termwise and modulo $k$) to every codeword of $\mathcal{G}$, and then appending the suffix $\vec{a}$ to the resulting $n$-tuple. In symbols,

$$\{\mathcal{G} \oplus \vec{x}\} \otimes \vec{a} = \{z_1 \cdots z_n a_1 \cdots a_s \mid z_i \equiv y_i + x_i \bmod k \text{ for some } y_1 \cdots y_n \in \mathcal{G}, \text{ with } z_1, \ldots, z_n \in \{0, 1, \ldots, k-1\}\}.$$

Let us now define the code $\mathcal{D}$ by

$$\mathcal{D} = \bigcup_{i=0}^{k-1} \{\mathcal{C} \oplus i \underbrace{00 \cdots 0}_{n-1 \text{ times}}\} \otimes ii. \tag{7}$$

We claim that $\mathcal{D}$ satisfies the requirements of the lemma. By definition, the length of $\mathcal{D}$ is $n+2$. Since for all $0 \le i < j \le k-1$,

$$\{\mathcal{C} \oplus i00 \cdots 0\} \otimes ii \cap \{\mathcal{C} \oplus j00 \cdots 0\} \otimes jj = \varnothing,$$

we immediately obtain $| \mathcal{D} | = k \times | \mathcal{C} |$.

We shall now show that $\delta(\mathcal{D}) \ge 3$. Indeed, any two distinct codewords $\vec{x}, \vec{y} \in \mathcal{D}$ have the form $\vec{x} = \{\vec{x}' \oplus i00 \cdots 0\} \otimes ii$ and $\vec{y} = \{\vec{y}' \oplus j00 \cdots 0\} \otimes jj$ for suitable codewords $\vec{x}', \vec{y}' \in \mathcal{C}$ and $i, j \in \{0, 1, \ldots, k-1\}$.

We now argue by cases:

(i) If $i = j$ then $\vec{x}' \ne \vec{y}'$, whence

$$d_H(\vec{x}, \vec{y}) = d_H(\{\vec{x}' \oplus i00 \cdots 0\} \otimes ii, \{\vec{y}' \oplus i00 \cdots 0\} \otimes ii) = d_H(\vec{x}', \vec{y}') \ge 3,$$

by our hypothesis on $\delta(\mathcal{C})$.

(ii) If $i \ne j$ then

$$\begin{aligned} d_H(\vec{x}, \vec{y}) &= d_H(\{\vec{x}' \oplus i00 \cdots 0\} \otimes ii, \{\vec{y}' \oplus j00 \cdots 0\} \otimes jj) \\ &= d_H(\{\vec{x}' \oplus i00 \cdots 0\}, \{\vec{y}' \oplus j00 \cdots 0\}) + 2. \end{aligned}$$

If $\vec{x}' = \vec{y}'$ then $d_H(\{\vec{x}' \oplus i00 \cdots 0\}, \{\vec{y}' \oplus j00 \cdots 0\}) = 1$; hence $d_H(\vec{x}, \vec{y}) = 3$. If $\vec{x}' \ne \vec{y}'$ then $d_H(\{\vec{x}' \oplus i00 \cdots 0\}, \{\vec{y}' \oplus j00 \cdots 0\}) \ge d - 1 \ge 2$, whence $d_H(\vec{x}, \vec{y}) \ge 4$.

Finally, by definition $\mu(\mathcal{D}) = \mu(\mathcal{C}) \ge e$. The proof is complete. ∎

The following lemma directly follows from the well known Gilbert bound [13].

LEMMA 3.3. *Let $n \geq 0, k \geq 2$. Let $M \geq 0$ be an integer satisfying the inequality*

$$M \leq \frac{k^n - \sum_{i=0}^{3} \binom{n}{i}(k-1)^i}{\sum_{i=0}^{2} \binom{n}{i}(k-1)^i}.$$

*Then there exists an $(n, M, 3)$ $k$-ary code $\mathscr{C}$ with $\mu(\mathscr{C}) \geq 4$.*

*Proof.* We can safely identify our alphabet with the set $\mathscr{K} = \{0, 1, \ldots, k-1\}$. Let $\mathscr{C}'$ be the largest $k$-ary code of length $n$ such that $\delta(\mathscr{C}') \geq 3$ and $\mu(\mathscr{C}') \geq 4$. Then there is no word in $\mathscr{K}^n$ simultaneously having distance $\geq 3$ from each word in $\mathscr{C}'$, and distance $\geq 4$ from $\vec{0}$. Stated otherwise, the spheres $\mathscr{B}_2(\vec{c})$, with $\vec{c} \in \mathscr{C}'$, cover $\mathscr{K}^n \setminus \mathscr{B}_3(\vec{0})$. We then conclude that the sum $|\mathscr{C}'| \sum_{i=0}^{2} \binom{n}{i}(k-1)^i$ of the volumes of these spheres is $\geq |\mathscr{K}^n \setminus \mathscr{B}_3(\vec{0})| = k^n - \sum_{i=0}^{3} \binom{n}{i}(k-1)^i$. ∎

## 4. OPTIMAL STRATEGIES FOR BINARY SEARCH WITH MINIMUM ADAPTIVENESS

By Lemma 2.1(ii), at least $\mathrm{ch}(2^m, 0, 0)$ questions are *necessary* for Paul to guess the unknown number $x_{\text{Carole}} \in S = \{0, 1, \ldots, 2^m - 1\}$, if up to two answers may be erroneous. In this section we shall prove that, conversely (with the exception of $m = 2$ and $m = 4$), $\mathrm{ch}(2^m, 0, 0)$ questions are *sufficient* under the following constraint: Paul first sends to Carole a batch of $m$ non-adaptive questions $D_1, \ldots, D_m$, and then, only depending on Carole's answers, he sends $\mathrm{ch}(2^m, 0, 0) - m$ non-adaptive questions in a second batch. More precisely, the first batch of questions asks for the binary representation of $x_{\text{Carole}}$. The above *perfect* strategy is "canonical" in the following sense

DEFINITION 4.1. A strategy $\mathscr{S}$ for a state $\sigma$ of type $(2^m, 0, 0)$ is said to be *canonical* iff $\mathscr{S}$ is winning for $\sigma$ and consists of two batches of non-adaptive questions, where the questions in the first batch ask for the binary digits of $x_{\text{Carole}}$, and the second batch only depends on the $m$-tuple of Carole's answers to these questions.

In Lemma 4.11 below we shall see that a *perfect* strategy with minimum adaptiveness, albeit *non-canonical*, also exists for the case $m = 4$.

### *Canonical Binary Search with Minimum Adaptiveness*

The *first batch of questions* is described as follows:

> For each $i = 1, 2, \ldots, m$, let $D_i \subseteq S$ denote the question "Is the $i$th binary digit of $x_{\text{Carole}}$ equal to 1?" Thus a number $y \in S$ belongs to $D_i$ iff the $i$th bit $y_i$ of its binary expansion $\vec{y} = y_1 \cdots y_m$ is equal to 1.

Upon identifying 1 = "yes" and 0 = "no," let $b_i \in \{0, 1\}$ be Carole's answer to question $D_i$. Let $\vec{b} = b_1 \cdots b_m$. Repeated application of (1), (2), beginning with the initial state $\sigma = (S, \varnothing, \varnothing)$, shows that Paul's state of knowledge as an effect of Carole's answers is a triplet $\sigma^{\vec{b}} = (A_0, A_1, A_2)$, where

$A_0$ = the singleton containing the number whose binary expansion equals $\vec{b}$

$A_1 = \{y \in S \mid d_H(\vec{y}, \vec{b}) = 1\}$

$A_2 = \{y \in S \mid d_H(\vec{y}, \vec{b}) = 2\}$.

By direct verification we have $|A_0| = 1$, $|A_1| = m$, $|A_2| = \binom{m}{2}$. Thus the state $\sigma^{\vec{b}}$ is of type $(1, m, \binom{m}{2})$. As in (3), let $\sigma_i$ be the state resulting after Carole's first $i$ answers, beginning with $\sigma_0 = \sigma$. Since each question $D_i$ is balanced for $\sigma_{i-1}$, an easy induction using Lemma 2.3 yields $\mathrm{ch}(\sigma^{\vec{b}}) = \mathrm{ch}(2^m, 0, 0) - m$.

*The Critical Index $m_n$*

For each $m$-tuple $\vec{b} \in \{0, 1\}^m$ of Carole's answers, we shall construct a non-adaptive strategy with $\mathrm{ch}(1, m, \binom{m}{2})$ questions, which turns out to be winning for the state $\sigma^{\vec{b}}$. To this purpose, let us consider the values of $\mathrm{ch}(1, m, \binom{m}{2})$ for $m \geq 1$. A direct computation yields $\mathrm{ch}(1, 1, 0) = 4$, $\mathrm{ch}(1, 2, 1) = 5$, $\mathrm{ch}(1, 3, 3) = \mathrm{ch}(1, 4, 6) = 6$, $\mathrm{ch}(1, 5, 10) = \cdots = \mathrm{ch}(1, 8, 28) = 7$, $\mathrm{ch}(1, 9, 36) = \cdots = \mathrm{ch}(1, 14, 91) = 8, \ldots$ .

DEFINITION 4.2. Let $n \geq 4$ be an arbitrary integer. The *critical index* $m_n$ is the largest integer $m \geq 0$ such that $\mathrm{ch}(1, m, \binom{m}{2}) = n$. Thus,

$$\mathrm{ch}\left(1, m_n, \binom{m_n}{2}\right) = n \qquad \text{and} \qquad \mathrm{ch}\left(1, m_n + 1, \binom{m_n + 1}{2}\right) > n. \quad (8)$$

The function $n \mapsto m_n$ is well defined for all $n \geq 4$. The first values of $m_n$ are given by

$$m_4 = 1, \quad m_5 = 2, \quad m_6 = 4, \quad m_7 = 8, \quad (9)$$
$$m_8 = 14, \quad m_9 = 22, \quad m_{10} = 34,$$

$$m_{11} = 52, \quad m_{12} = 78, \quad m_{13} = 114, \quad (10)$$
$$m_{14} = 166, \quad m_{15} = 240, \ldots .$$

As usual, for every real number $\rho$ we denote by $\lfloor \rho \rfloor$ the largest integer $k \leq \rho$.

LEMMA 4.3. *Let $n \geq 4$ be an arbitrary integer.*

(i) *If $n$ is odd then $m_n = 2^{(n+1)/2} - n - 1$.*

(ii) *If $n$ is even then, letting $m^* = \lfloor 2^{(n+1)/2} \rfloor - n - 1$, we either have $m_n = m^*$ or $m_n = m^* + 1$.*

*Proof.* The case $n = 4$ is settled by direct verification, recalling that $m_4 = 1$. For the case $n \geq 5$ see [14, Lemma 4.2], where our present $m_n$ is denoted $n_\chi$ and is called the *first* critical index. ∎

*Strategies and Codes*: *The Second Batch of Questions*

As a key tool for the construction of the second batch of questions we prepare the following

LEMMA 4.4. *For any integers $a_0, a_1, a_2 \geq 0$, let $\sigma = (A_0, A_1, A_2)$ be a state of type $(a_0, a_1, a_2)$ and $n = \mathrm{ch}(a_0, a_1, a_2)$. Then a non-adaptive winning strategy for $\sigma$ with $n$ questions exists* if and only if *for some integers $d_0 \geq 5$ and $d_1 \geq 3$ there exist an $(n, a_0, d_0)$ binary code $\mathscr{C}_0$ and an $(n, a_1, d_1)$ binary code $\mathscr{C}_1$ such that $\Delta(\mathscr{C}_0, \mathscr{C}_1) \geq 4$.*

*Proof.* $\Rightarrow$ Assume $\sigma = (A_0, A_1, A_2)$ to be a state of type $(a_0, a_1, a_2)$ having a non-adaptive winning strategy $\mathscr{S}$ with $n$ questions $T_1, \ldots, T_n$. Let the map

$$z \in A_0 \cup A_1 \cup A_2 \mapsto \vec{z}^{\mathscr{S}} \in \{0,1\}^n$$

send each $z \in A_0 \cup A_1 \cup A_2$ into the $n$-tuple of bits $\vec{z}^{\mathscr{S}} = z_1^{\mathscr{S}} \cdots z_n^{\mathscr{S}}$ arising, via the identifications "yes" $= 1$ and "no" $= 0$, from the answers to the questions "does $z$ belong to $T_1$?," "does $z$ belong to $T_2$?," ..., "does $z$ belong to $T_n$?" More precisely, for each $j = 1, \ldots, n$, $z_j^{\mathscr{S}} = 1$ iff $z \in T_j$. Let $\mathscr{C} \subseteq \{0,1\}^n$ be the range of the map $z \mapsto \vec{z}^{\mathscr{S}}$.

We shall prove that, for each $i \in \{0,1\}$, the set $\mathscr{C}_i = \{\vec{y}^{\mathscr{S}} \in \mathscr{C} \mid y \in A_i\}$ is an $(n, a_i, d_i)$ binary code for some $d_i \geq 5 - 2i$; further, for every $z \in A_1$ and $h \in A_0$ we shall establish the inequality $d_H(\vec{z}^{\mathscr{S}}, \vec{h}^{\mathscr{S}}) \geq 4$, i.e., $\Delta(\mathscr{C}_1, \mathscr{C}_2) \geq 4$.

Since $\mathscr{S}$ is winning, the map $z \mapsto \vec{z}^{\mathscr{S}}$ is one-one, whence in particular $|\mathscr{C}_0| = a_0$ and $|\mathscr{C}_1| = a_1$. Moreover, by definition, $\mathscr{C}_0$ and $\mathscr{C}_1$ are subsets of $\{0,1\}^n$. The remaining desired properties $\delta(\mathscr{C}_i) \geq 5 - 2i$ and $\Delta(\mathscr{C}_0, \mathscr{C}_1) \geq 4$ are direct consequences of the following

*Claim.* For all $i, j \in \{0,1\}$ and $\vec{c} \in \mathscr{C}_i$, $\vec{d} \in \mathscr{C}_j$ we have $d_H(\vec{c}, \vec{d}) \geq 4 - (i + j)$.

For otherwise (absurdum hypothesis) assuming $c \in A_i$ and $d \in A_j$ to be two distinct elements satisfying $d_H(\vec{c}^{\mathscr{S}}, \vec{d}^{\mathscr{S}}) < 4 - (i + j)$, we will prove that $\mathscr{S}$ is not a winning strategy. We can safely assume $c_k^{\mathscr{S}} = d_k^{\mathscr{S}}$ for each $k = 1, \ldots, n - 3 + (i + j)$. Suppose Carole's answer to question $T_k$ is "yes"

or "no" according as $c_k^{\mathscr{S}} = 1$ or $\not\in c_k^{\mathscr{S}} = 0$, respectively. Then $c$ and $d$ satisfy all of Carole's $n - 3 + (i + j)$ answers. It follows that Paul's resulting state of knowledge has the form $\sigma' = (A_0', A_1', A_2')$, with $c \in A_i'$ and $d \in A_j'$, whence the type of $\sigma'$ is $(a_0', a_1', a_2')$ with $a_i' + a_j' \geq 2$. By [3, Lemma 2.5] we have $\mathrm{ch}(\sigma') \geq 4 - (i + j)$. By Lemma 2.1(ii), in either case $3 - (i + j)$ questions/answers will not suffice to reach a final state, a contradiction.

$\Leftarrow$ Let $\mathscr{C}_0$ be an $(n, a_0, d_0)$ code, with $d_0 \geq 5$, and let $\mathscr{C}_1$ be an $(n, a_1, d_1)$ code, with $d_1 \geq 3$. Moreover, assume $\Delta(\mathscr{C}_0, \mathscr{C}_1) \geq 4$. Let $\mathscr{H} \subseteq \{0, 1\}^n$ be the union of the Hamming spheres of radius 2 centered at the codewords of $\mathscr{C}_0$, together with the Hamming spheres of radius 1 centered at the codewords of $\mathscr{C}_1$; in symbols, $\mathscr{H} = \bigcup_{\vec{x} \in \mathscr{C}_0} \mathscr{B}_2(\vec{x}) \cup \bigcup_{\vec{y} \in \mathscr{C}_1} \mathscr{B}_1(\vec{y})$. By our standing hypothesis on $\delta(\mathscr{C}_0)$, $\delta(\mathscr{C}_1)$ and $\Delta(\mathscr{C}_0, \mathscr{C}_1)$, it is not hard to see that, for any two distinct codewords $\vec{x}_0, \vec{x}_1 \in \mathscr{C}_0$ and any two distinct codewords $\vec{y}_0, \vec{y}_1 \in \mathscr{C}_1$, the Hamming spheres $\mathscr{B}_2(\vec{x}_0), \mathscr{B}_2(\vec{x}_1), \mathscr{B}_1(\vec{y}_0), B_1(\vec{y}_1)$ are pairwise disjoint. From (6) it follows that $|\mathscr{H}| = (\binom{n}{2} + n + 1)a_0 + (n + 1)a_1$. Let $\mathscr{C}_2 = \{0, 1\}^n \setminus \mathscr{H}$. Since $n = \mathrm{ch}(a_0, a_1, a_2)$, by definition of character we have $2^n \geq (\binom{n}{2} + n + 1)a_0 + (n + 1)a_1 + a_2$. It follows that $|\mathscr{C}_2| = 2^n - |\mathscr{H}| \geq a_2$. Trivially, $\mathscr{C}_2$ is an $(n, \chi, 1)$ binary code for some $\chi \geq a_2$. Let $\sigma = (A_0, A_1, A_2)$ be an arbitrary state of type $(a_0, a_1, a_2)$. Let us now fix, once and for all, three one-one maps $f_0$: $A_0 \to \mathscr{C}_0$, $f_1$: $A_1 \to \mathscr{C}_1$, and $f_2$: $A_2 \to \mathscr{C}_2$. The existence of $f_0, f_1$, and $f_2$ is ensured by the minimum distance of the the codes $\mathscr{C}_0, \mathscr{C}_1$, and $\mathscr{C}_2$.

Let the map $f$: $A_0 \cup A_1 \cup A_2 \to \{0, 1\}^n$ be defined by cases as follows:

$$f(y) = \begin{cases} f_0(y), & y \in A_0 \\ f_1(y), & y \in A_1 \\ f_2(y), & y \in A_2. \end{cases} \tag{11}$$

Note that $f$ is one-one. For each $y \in A_0 \cup A_1 \cup A_2$ and $j = 1, \ldots, n$ let $f(y)_j$ be the $j$th bit of the binary vector corresponding to $y$ via $f$.

> Let the set $T_j \subseteq S$ be defined by $T_j = \{z \in S \mid f(z)_j = 1\}$, $(j = 1, \ldots, n)$. This is Paul's *second batch of questions*. Intuitively, letting $x_{\text{Carole}}$ denote Carole's secret number, $T_j$ asks "is the $j$th bit of $f(x_{\text{Carole}})$ equal to 1?"

We shall show that the sequence $T_1, \ldots, T_n$ yields a perfect non-adaptive winning strategy for $\sigma$. Again writing "yes" = 1 and "no" = 0, Carole's answers to questions $T_1, \ldots, T_n$ determine an $n$-tuple of bits $\vec{a} = a_1 \cdots a_n$. As in (3), let $\sigma_i$ be the state resulting after Carole's first $i$ answers, beginning with $\sigma_0 = \sigma$. Arguing by cases, we shall show that $\sigma_n = (A_0^*, A_1^*, A_2^*)$ is a final state.

By (1), (2), for all $i = 0, 1, 2$, any $z \in A_i$ that falsifies $> 2 - i$ answers does not survive in $\sigma_n$—in the sense that $z \notin A_0^* \cup A_1^* \cup A_2^*$.

*Case 1.* $\vec{a} \notin \bigcup_{h \in A_0} \mathscr{B}_2(f(h)) \cup \bigcup_{y \in A_1} \mathscr{B}_1(f(y)) \cup f(A_2)$.

Then for any $i = 0, 1, 2$ and for each $h \in A_i$ we have $h \notin A_0^* \cup A_1^* \cup A_2^*$. As a matter of fact, from $\vec{a} \notin \mathscr{B}_{2-i}(f(h))$, it follows that $d_H(f(h), \vec{a}) > 2 - i$, whence $h$ falsifies $> 2 - i$ of the answers to $T_1, \ldots, T_n$, and $h$ does not survive in $\sigma_n$. We have proved that $A_0^* \cup A_1^* \cup A_2^*$ is empty, and $\sigma_n$ is a final state.

*Case 2.* $\vec{a} \in \mathscr{B}_{2-i}(f(h))$ for some $i \in \{0, 1, 2\}$ and $h \in A_i$.

Then $h \in A_0^* \cup A_1^* \cup A_2^*$, because $d_H(f(h), \vec{a}) \leq 2 - i$, whence $h$ falsifies $\leq 2 - i$ answers. Our assumptions about $\mathscr{C}_0$, $\mathscr{C}_1$, and $\mathscr{C}_2$ ensure that, for all $j = 0, 1, 2$ and each $h' \in A_j$ (with $h' \neq h$) we have $\vec{a} \notin \mathscr{B}_{2-j}(f(h'))$. Thus, $d_H(f(h'), \vec{a}) > 2 - j$ and $h'$ falsifies $> 2 - j$ of the answers to $T_1, \ldots, T_n$, whence $h'$ does not survive in $\sigma_n$. This shows that $h' \notin A_0^* \cup A_1^* \cup A_2^*$. Therefore, $A_0^* \cup A_1^* \cup A_2^*$ only contains the element $h$, and $\sigma_n$ is a final state. ∎

COROLLARY 4.5. *Let* $m = 1, 2, 3, \ldots$ *and* $n = \mathrm{ch}(1, m, \binom{m}{2})$. *Let* $\sigma = (A_0, A_1, A_2)$ *be any state of type* $(1, m, \binom{m}{2})$. *Then there exists a non-adaptive winning strategy for* $\sigma$ *with* $n$ *questions* if and only if *for some integer* $d \geq 3$ *there exists an* $(n, m, d)$ *binary code with minimum Hamming weight* $\geq 4$.

*Proof.* If there exists a non-adaptive winning strategy for $\sigma$ with $n$ questions then by Lemma 4.4 there exist an $(n, 1, d_0)$ code $\mathscr{C}_0$ and an $(n, m, d_1)$ code $\mathscr{C}_1$ with $d_0 \geq 5$, $d_1 \geq 3$, and $\Delta(\mathscr{C}_0, \mathscr{C}_1) \geq 4$. Let $\mathscr{C}_0 = \{\vec{h}\}$. Let

$$\mathscr{C} = \{\vec{y} \oplus \vec{h} \mid \vec{y} \in \mathscr{C}_1\},$$

where $\oplus$ stands for bitwise sum modulo 2. For any two distinct codewords $\vec{a}, \vec{b} \in \mathscr{C}$ we have $\vec{a} = \vec{c} \oplus \vec{h}$ and $\vec{b} = \vec{d} \oplus \vec{h}$, for uniquely determined elements $\vec{c}, \vec{d} \in \mathscr{C}_1$. Thus we get $d_H(\vec{a}, \vec{b}) = d_H(\vec{c} \oplus \vec{h}, \vec{d} \oplus \vec{h}) = d_H(\vec{c}, \vec{d}) \geq d_1 \geq 3$, whence $\delta(\mathscr{C}) \geq 3$. Using the abbreviation

$$\vec{0} = \underbrace{0 \cdots 0}_{n \text{ times}},$$

we have $w_H(\vec{a}) = d_H(\vec{a}, \vec{0}) = d_H(\vec{c} \oplus \vec{h}, \vec{h} \oplus \vec{h}) = d_H(\vec{c}, \vec{h}) \geq 4$, whence $\mu(\mathscr{C}) \geq 4$. In conclusion, $\mathscr{C}$ is an $(n, m, d)$ code with $d \geq 3$ and $\mu(\mathscr{C}) \geq 4$, as required.

Conversely, let $\mathscr{C}$ be an $(n, m, d)$ code with $d \geq 3$ and $\mu(\mathscr{C}) \geq 4$. Let $\mathscr{D} = \{\vec{0}\}$. Then $\mathscr{D}$ is an $(n, 1, d')$ code for every $d' \geq 1$. Furthermore, we have $\Delta(\mathscr{C}, \mathscr{D}) \geq 4$. Thus by Lemma 4.4 there exists a non-adaptive winning strategy for $\sigma$ with $n$ questions. The proof is complete. ∎

LEMMA 4.6. *For each integer* $n \geq 7$ *there exists an integer* $d \geq 3$ *and an* $(n, m_n, d)$ *code* $\mathscr{C}_n$ *such that* $\mu(\mathscr{C}_n) \geq 4$, *where* $m_n$ *is as in Definition* 4.2.

*Proof.* We shall argue by cases.

*Case 1.* $7 \leq n < 11$.

With reference to (9), (10) we can write $m_7 = 8, m_8 = 14, m_9 = 22, m_{10} = 34$. By direct inspection in [5, Table I-A], for suitable integers $e_1, e_2 > 0$ with $e_1 + e_2 \geq m_n$ there exist an $(n, e_1, 4)$ binary code $\mathscr{A}_1$ and an $(n, e_2, 4)$ binary code $\mathscr{A}_2$. Moreover, for every $\vec{x} \in \mathscr{A}_1$, $w_H(\vec{x}) = 4$ and for all $\vec{y} \in \mathscr{A}_2$, $w_H(\vec{y}) = 7$; hence $d_H(\vec{x}, \vec{y}) \geq 3$. It follows that every set $\mathscr{C}_n \subseteq \mathscr{A}_1 \cup \mathscr{A}_2$ such that $|\mathscr{C}_n| = m_n$ is an $(n, m_n, 3)$ binary code with the additional property $\mu(\mathscr{C}_n) = 4$, as required.

*Case 2.* $n \geq 11$.

*Claim.* There exists an $(n, e, d)$ binary code $\mathscr{D}_n$ such that $e \geq 2^{(n+1)/2}$, $d \geq 3$, $\mu(\mathscr{D}_n) \geq 4$.

We argue by induction on $n$.

*Basis.* $(n = 11, 12)$. Then direct inspection in [5, Table I-A] yields two binary codes $\mathscr{A}_1, \mathscr{A}_2$, such that

- $\mathscr{A}_1$ is an $(11, 66, 4)$ code and for every $\vec{x} \in \mathscr{A}_1$, $w_H(\vec{x}) = 6$;
- $\mathscr{A}_2$ is a $(12, 132, 4)$ code and for every $\vec{x} \in \mathscr{A}_2$, $w_H(\vec{x}) = 6$.

Let $\mathscr{D}_{11} = \mathscr{A}_1$ and $\mathscr{D}_{12} = \mathscr{A}_2$. The inequalities $132 > 2^{13/2}$ and $66 > 2^6$ now settle our claim for $n \in \{11, 12\}$.

*Induction Step.* Assuming the claim to hold for $n \geq 11$, by Lemma 3.2 the claim also holds for $n + 2$, as required.

From Lemma 4.3 we get $2^{(n+1)/2} > m_n$, whence the desired conclusion now follows from our claim by arbitrarily picking a subcode $\mathscr{C}_n \subseteq \mathscr{D}_n$ with $|\mathscr{C}_n| = m_n$. ∎

COROLLARY 4.7. *For $m = 5, 6, 7, \ldots$ let $\sigma$ be a state of type $(1, m, \binom{m}{2})$. Then there exists a perfect strategy $\mathscr{S}$ for $\sigma$. In other words, $\mathscr{S}$ is winning for $\sigma$ and the number of questions in $\mathscr{S}$ coincides with Berlekamp's lower bound* $\mathrm{ch}(\sigma) = \mathrm{ch}(2^m, 0, 0) - m$.

*Proof.* Let $n = \mathrm{ch}(\sigma)$. From the assumption $m \geq 5$ we get $n \geq 7$. By Definition 4.2, $m \leq m_n$. By Lemma 4.6 there exists an $(n, m_n, d)$ binary code $\mathscr{C}_n$ with $d \geq 3$ and $\mu(\mathscr{C}_n) \geq 4$. Picking now a subcode $\mathscr{C}'_n \subseteq \mathscr{C}_n$ such that $|\mathscr{C}'_n| = m$ and applying Corollary 4.5 we have the desired conclusion. ∎

Turning our attention to the remaining cases, we shall prove that Corollary 4.7 also holds when $m = 1$ and $m = 3$. For $m = 2$ and $m = 4$ we shall prove that the shortest non-adaptive winning strategy for a state of type $(1, m, \binom{m}{2})$ requires $\mathrm{ch}(1, m, \binom{m}{2}) + 1$ questions.

LEMMA 4.8. *Let $\mathscr{C}$ be the largest binary code of length 6 such that $\mu(\mathscr{C}) = 4$ and $\delta(\mathscr{C}) \geq 3$. Then $|\mathscr{C}| = 3$.*

*Proof.* Since the code $\mathscr{D} = \{111100, 110011, 001111\}$ satisfies $\delta(\mathscr{D}) \geq 3$, $\mu(\mathscr{D}) = 4$, and $|\mathscr{D}| = 3$, then the largest code $\mathscr{C}$ satisfying the requirements of the lemma necessarily has $\geq 3$ elements.

Conversely, we shall prove that any such $\mathscr{C}$ has $\leq 3$ (and hence, exactly 3) elements. Let us write $\mathscr{C} = \mathscr{C}^{(4)} \cup \mathscr{C}^{(5)} \cup \mathscr{C}^{(6)}$, where $\mathscr{C}^{(i)} = \{\vec{x} \in \mathscr{C} \mid w_H(\vec{x}) = i\}$. We shall prove the following easy facts:

(a) $|\mathscr{C}^{(5)} \cup \mathscr{C}^{(6)}| \leq 1$.

(b) $|\mathscr{C}^{(4)}| \leq 3$.

(c) If $|\mathscr{C}^{(4)}| = 3$ then $|\mathscr{C}^{(5)} \cup \mathscr{C}^{(6)}| = 0$.

There cannot exist two distinct codewords $\vec{y}_1, \vec{y}_2 \in \mathscr{C}^{(5)} \cup \mathscr{C}^{(6)}$, for otherwise, $d_H(\vec{y}_1, \vec{y}_2) \leq 2$, against the hypothesis $\delta(\mathscr{C}) \geq 3$. This settles (a).

To prove (b), let $\vec{x}_1, \ldots, \vec{x}_n$ be the list of codewords of $\mathscr{C}^{(4)}$. For each $i = 1, \ldots, n$ let $N_i = \{\vec{y} \in \{0,1\}^6 \mid d_H(\vec{x}_i, \vec{y}) \leq 1 \text{ and } w_H(\vec{y}) = 5\}$. Each $N_i$ has exactly two elements, and whenever $i \neq j$ we have $N_i \cap N_j = \emptyset$. It follows that $|\bigcup_{i=1}^n N_i| = \Sigma_{i=1}^n |N_i| = 2n$. Therefore,

$$2n = |\bigcup_{i=1}^{n} N_i| \leq |\{\vec{x} \in \{0,1\}^6 \mid w_H(\vec{x}) = 5\}| = 6,$$

and $n \leq 3$, as desired.

Finally, to prove (c), assume $|\mathscr{C}^{(4)}| = 3$. Since by the above proof of (b), $\bigcup_{i=1}^n N_i$ exhausts the set of 6-tuples of bits having Hamming weight 5, every 6-tuple of bits having Hamming weight 5 is contained in the Hamming sphere of radius 1 centered at some codeword in $\mathscr{C}^{(4)}$. From the assumption $\delta(\mathscr{C}) \geq 3$ it follows that $|\mathscr{C}^{(5)}| = 0$. Finally, $\mathscr{C}^{(6)}$ must be empty, because its only element 111111 has Hamming distance 2 from every element of $\mathscr{C}^{(4)}$. ∎

PROPOSITION 4.9. *For each $m = 1, 2, 3, 4$ let $\lambda(m)$ be the length of the shortest non-adaptive winning strategy for some (equivalently, for every) state of type $(1, m, \binom{m}{2})$. Then*

$$\lambda(1) = 4, \qquad \lambda(2) = 6, \qquad \lambda(3) = 6, \qquad \lambda(4) = 7.$$

*For $m \in \{1, 3\}$, and only for such values of $m$, the number $\lambda(m)$ satisfies the condition $\lambda(m) = \mathrm{ch}(1, m, \binom{m}{2})$.*

*Proof.* For $m = 1$ we have $\lambda(1) \geq \mathrm{ch}(1, 1, 0) = 4$. Conversely, by Corollary 4.5, using the singleton code $\{1111\}$, we also get $\lambda(1) \leq 4$.

For $m = 2$, by [9, pp. 75–76], any winning strategy for a state of type $(1, 2, 1)$ necessarily uses $\geq 6$ questions—even in the fully interactive model,

where each question is adaptively asked after receiving the answer to the previous questions. A fortiori, in our present non-adaptive case, $\lambda(2) \geq 6$. On the other hand, taking the code $\mathscr{C} = \{111100, 001111\}$ and using Corollary 4.5, one obtains a non-adaptive strategy with six questions which is winning for every state of type $(1, 2, 1)$. Thus $\lambda(2) \leq 6$.

For $m = 3$ we have $\lambda(3) \geq \mathrm{ch}(1, 3, 3) = 6$. Conversely, combining Corollary 4.5 and Lemma 4.8 we get $\lambda(3) \leq 6$.

Finally, let us consider the case $m = 4$. On the one hand, Corollary 4.5 and Lemma 4.8 are to the effect that $\lambda(4) \geq 7$. On the other hand, taking the $(7, 4, 3)$ code $\mathscr{C} = \{1111000, 0001111, 0110011, 1111111\}$ and again using Corollary 4.5, we obtain a non-adaptive winning strategy with seven questions for any state of type $(1, 4, 6)$. Therefore, $\lambda(4) \leq 7$, and the proof is complete. ∎

Combining Corollary 4.7 and Proposition 4.9 we have

THEOREM 4.10. *For each integer $m = 1, 3, 5, 6, 7, 8, \ldots$ there is a binary searching strategy $\mathscr{S}$ to guess a number $x \in \{0, \ldots, 2^m - 1\}$ with up to two lies in the answers, which is perfect and canonical. Thus, $\mathscr{S}$ uses a first batch of $m$ non-adaptive questions asking for the bits of the binary expansion of $x$ and then, only depending on the answers to these questions, a second batch of $\mathrm{ch}(2^m, 0, 0) - m$ non-adaptive questions.*

*In case $m \in \{2, 4\}$, let $\mathscr{S}$ be the shortest canonical strategy to guess a number $x \in \{0, \ldots, 2^m - 1\}$ with up to two lies in the answers. Then $\mathscr{S}$ requires precisely $\mathrm{ch}(2^m, 0, 0) + 1$ questions.*

*A Non-canonical Perfect Strategy for the Case $m = 4$*

The above theorem leaves open the possibility that there exist perfect non-canonical strategies for the exceptional cases $m = 2$ and $m = 4$. The following lemma shows that this is indeed the case for $m = 4$. A final remark in this section will (negatively) take care of the case $m = 2$.

LEMMA 4.11. *There exists a* perfect *binary strategy to guess a number $x \in S = \{0, \ldots, 15\}$ with up to two lies in the answers, using a first batch of five non-adaptive questions and then, only depending on the answers to these questions, a second batch of five non-adaptive questions.*

*Proof.* Let $x_{\text{Carole}}$ denote Carole's secret number. We can safely identify each $x \in S$ with the four bit string $x_1x_2x_3x_4$ yielding the binary expansion of $x$. Paul's first batch $[Q_1, \ldots, Q_5]$ of non-adaptive questions is as follows: For each $i = 1, \ldots, 4$, question $Q_i$ asks

> "Is the $i$th bit of (the binary expansion of) $x_{\text{Carole}}$ equal to 1?"

Question $Q_5$ asks

> "Is the sum modulo 2 of the first three bits of $x_{\text{Carole}}$ equal to 1?"

Let $\sigma = (A_0, A_1, A_2)$ denote Paul's state resulting from Carole's answers to questions $Q_1, \ldots, Q_4$. There is precisely one element $h = h_1h_2h_3h_4 \in S$ such that $A_0 = \{h\}$. Specifically, using the identifications "yes" = 1 and "no" = 0, the $i$th bit $h_i$ of the only element of $A_0$ coincides with Carole's answer to the $i$th question. Each element $x = x_1x_2x_3x_4 \in A_1$ has precisely one *discrepancy* from $h$ (in the sense that $x_j = h_j$ for all $j = 1, 2, 3, 4$ except one.) Each element $y = y_1y_2y_3y_4 \in A_2$ has exactly two discrepancies from $h$. Direct inspection shows that the type of $\sigma$ is $(1, 4, 6)$. Let $\sigma^{\surd} = (A_0^{\surd}, A_1^{\surd}, A_2^{\surd})$ be the state resulting from the first five answers. Then, denoting by $b \in \{0, 1\}$ Carole's answer to the fifth question, the state $\sigma^{\surd}$ arises from $\sigma$ in accordance to the formation rules (1), (2).

*Claim 1.* The type of $\sigma^{\surd}$ is either $(1, 1, 6)$ or $(0, 4, 4)$.

We shall argue by cases as follows.

*Case 1.* $h$ satisfies Carole's fifth answer; i.e., $h_1 + h_2 + h_3 \equiv b \bmod 2$.

Then $A_0^{\surd} = A_0 = \{h\}$. An element $x = x_1x_2x_3x_4 \in A_1$ satisfies Carole's fifth answer iff its unique discrepancy from $h$ occurs in the *fourth* position—so that the sum modulo 2 of its first three bits is the same as for $h$. Only one element $x^* \in A_1$ satisfies this condition, and $A_1^{\surd}$ will only consist of this element. The three remaining elements of $A_1$ will survive as elements of $A_2^{\surd}$, because they falsify exactly two of Carole's answers to $Q_1, \ldots, Q_5$. An element $y = y_1y_2y_3y_4 \in A_2$ satisfies Carole's fifth answer iff its two discrepancies from $h$ both occur in the first three positions—so that the sum modulo 2 of the first three bits of $y$ is the same as for $h$. The three elements in $A_2$ satisfying this condition will survive in $A_2^{\surd}$, together with the three elements of $A_1$ other than $x^*$. Since the remaining elements of $A_2$ falsify three answers, they do not survive in $\sigma^{\surd}$. For the case under consideration, we have proved that $\sigma^{\surd}$ is of type $(1, 1, 6)$.

*Case 2.* $h$ falsifies Carole's fifth answer; i.e., $h_1 + h_2 + h_3 \equiv 1 - b \bmod 2$.

Then no element of $S$ satisfies all five answers, and $A_0^{\surd} = \varnothing$. Since $h$ only falsifies Carole's answer to $Q_5$ then $h \in A_1^{\surd}$. An element $x = x_1x_2x_3x_4 \in A_1$ belongs to $A_1^{\surd}$ iff it satisfies the fifth answer, iff its unique discrepancy from $h$ occurs among its first three bits—so that the sum of these three bits modulo 2 coincides with Carole's answer. The three elements in $A_1$ satisfying this condition will be members of $A_1^{\surd}$, together with $h$. The remaining element $x^\dagger \in A_1$ will survive in $A_2^{\surd}$. An element $y = y_1y_2y_3y_4 \in A_2$ belongs to $A_2^{\surd}$ iff it satisfies Carole's fifth answer iff its two discrepancies from $h$ are not both occurring in the first three positions: this latter condition is necessary and sufficient for $y_1 + y_2 + y_3 \equiv b \equiv 1 - (h_1 + h_2 + h_3) \bmod 2$. The three elements in $A_2$ satisfying this condition belong to $A_2^{\surd}$, together with

$x^\dagger$. The remaining members of $A_2$ do not survive in $\sigma^\surd$. We have proved that in the present case, $\sigma^\surd$ is of type $(0, 4, 4)$, and the claim is proved.

*Claim 2.* For any state $\sigma^\surd$ of either type $(1, 1, 6)$ or $(0, 4, 4)$ Paul has a non-adaptive winning strategy with five questions.

Indeed, if $\sigma^\surd$ is of type $(1, 1, 6)$ then $\mathrm{ch}(\sigma^\surd) = 5$. Let $\mathscr{C}_0 = \{00000\}$ and $\mathscr{C}_1 = \{11111\}$. Then $\Delta(\mathscr{C}_0, \mathscr{C}_1) = 5$. Further, for all integers $d_0 \geq 5$ and $d_1 \geq 3$, $\mathscr{C}_i$ is a $(5, 1, d_i)$ binary code (for each $i \in \{0, 1\}$). By Lemma 4.4 there exists a non-adaptive winning strategy for $\sigma^\surd$ with five questions.

If, on the other hand, $\sigma^\surd$ is of type $(0, 4, 4)$ then, again, $\mathrm{ch}(\sigma^\surd) = 5$. Let $\mathscr{C}_0 = \varnothing$ and $\mathscr{C}_1 = \{11100, 10111, 01011, 00000\}$. According to Definition 3.1, we can write $\Delta(\mathscr{C}_0, \mathscr{C}_1) \geq 4$. In the same way, $\mathscr{C}_0$ is a $(5, 0, d)$ binary code for every integer $d \geq 5$, and $\mathscr{C}_1$ is a $(5, 4, 3)$ binary code. By Lemma 4.4 there exists a non-adaptive winning strategy for $\sigma^\surd$ with five questions. Our second claim is settled.

Since $\mathrm{ch}(2^4, 0, 0) = 10$, our two claims yield the desired perfect strategy. ∎

*Remark.* As proved in [9], *in the fully adaptive game* with two lies, a perfect strategy exists to find an $m$ bit number iff $m \neq 2$. Therefore, combining Theorem 4.10 and the above Lemma 4.11 we have now the stronger result that *even if Paul is allowed to adapt his strategy only once, i.e., in the game with one-shot feedback, a perfect strategy exists to find an m bit number with two lies iff* $m \neq 2$.

## 5. EXTENSIONS TO $k$-ARY SEARCH

We shall now consider the Ulam–Rényi problem with two lies and $k$-ary search. In the corresponding game one now assumes that Paul asks questions having $k$ distinct possible answers.

We shall generalize Theorem 4.10 by proving that, for any fixed integer $k = 2, 3, 4, \ldots$, with finitely many exceptions $m$, Paul can find Carole's unknown number $x_{\text{Carole}}$ in the set $\{0, 1, \ldots, k^m - 1\}$ using Berlekamp's minimum number of $k$-ary questions and being allowed to adapt his strategy only once. Furthermore, we shall prove that for all $k = 2, 3, \ldots$, and $m = 1, 2, \ldots$, an *optimal* $k$-ary search can be achieved by strategies using minimum adaptiveness. This strengthens the results in [8, 9].

*Notation.* In the Ulam–Rényi game with two lies and $k$-ary search, Paul and Carole first fix two integers $k \geq 2$ and $m \geq 1$. The search space $S$ is identified with the set $\{0, 1, \ldots, k^m - 1\}$. The definition of *state* and *final state* are the same as in Section 2. Typically, a $k$-ary *question* $\mathbf{T}$ has the form

"Which one of the sets $T_0, T_1, \ldots, T_{k-1}$ does $x_{\text{Carole}}$ belong to?,"

where $\mathbf{T} = (T_0, T_1, \ldots, T_{k-1})$ is a $k$-tuple of (possibly empty) pairwise disjoint subsets of $S$ whose union is $S$.[5] Carole's answer is an integer $i \in \{0, 1, \ldots, k-1\}$, telling Paul that $x_{\text{Carole}}$ belongs to $T_i$. Generalizing (1), (2), if Paul is in state $\sigma = (A_0, A_1, A_2)$ and Carole's answer is equal to $i$, then Paul's state becomes

$$\sigma^i = (A_0 \cap T_i, (A_0 \setminus T_i) \cup (A_1 \cap T_i), (A_1 \setminus T_i) \cup (A_2 \cap T_i)). \tag{12}$$

A $k$-ary *strategy* with $q$ questions is a $k$-ary tree of depth $q$ where each node $\nu$ is labelled by a $k$-ary question $\mathbf{T}_\nu$. The definitions of *winning* and *non-adaptive* $k$-ary strategies are the natural generalizations of those given in Section 2.

For every integer $k \geq 2$ and state $\sigma$ of type $(a_0, a_1, a_2)$, *Berlekamp's (k-ary) weight of $\sigma$ before $q$ questions* is defined by

$$w_q^{[k]}(\sigma) = a_0 \left( \binom{q}{2} (k-1)^2 + q(k-1) + 1 \right) + a_1(q(k-1)+1) + a_2. \tag{13}$$

This is a generalization of (4). Accordingly, Lemma 2.1 has the following $k$-ary generalization.

PROPOSITION 5.1. *Let $\sigma$ be an arbitrary state and let $\mathbf{T}$ be a question. Define* $\mathrm{ch}^{[k]}(\sigma) = \min\{q = 0, 1, 2, \ldots \mid w_q^{[k]}(\sigma) \leq k^q\}$. *Let $\sigma^i$ be as in* (12).

(i) *For every integer $q \geq 1$ we have*

$$w_q^{[k]}(\sigma) = \sum_{i=1}^{k} w_{q-1}^{[k]}(\sigma^i).$$

(ii) *If $\sigma$ has a winning $k$-ary strategy with $q$ questions then $q \geq \mathrm{ch}^{[k]}(\sigma)$.*

See [1, 8] for a detailed discussion of the properties of $w_q^{[k]}$.

Generalizing Definition 2.2, by a *perfect* $k$-ary strategy for $\sigma$ we now mean a winning strategy for $\sigma$ only requiring $\mathrm{ch}^{[k]}(\sigma)$ questions. Generalizing Definition 4.1, we say that a strategy $\mathcal{S}$ for a state $\sigma$ of type $(k^m, 0, 0)$ is *canonical* iff $\mathcal{S}$ is winning for $\sigma$ and consists of two batches of non-adaptive questions, where the questions in the first batch ask for the $k$-ary digits of $x_{\text{Carole}}$, and the second batch only depends on the $m$-tuple of Carole's answers to these questions.

[5] Whenever Paul's state of knowledge $\sigma = (A_0, A_1, A_2)$ is clear from the context, it will be tacitly assumed that a question actually partitions only the set $A_0 \cup A_1 \cup A_2$ of surviving elements in $\sigma$. (If so desired, for the sake of definiteness, the remaining elements of $S$ can be safely attached to $T_{k-1}$.)

*Canonical k-ary Strategies with Minimum Adaptiveness*

To guess the secret number $x_{\text{Carole}}$ in $\text{ch}^{[k]}(k^m, 0, 0)$ questions, by analogy with the binary case, Paul adopts a canonical strategy $\mathcal{S}$ as follows: He first non-adaptively asks for the $k$-ary expansion of $x_{\text{Carole}}$—thus spending $m$ questions. After receiving Carole's answers, Paul fixes a $k$-ary encoding of the surviving candidates, and then non-adaptively asks Carole for the updated encoding of $x_{\text{Carole}}$. With finitely many exceptions $m$, the success of Paul's search is guaranteed by Theorem 5.9 below, which in turns relies on a multitude of results in the theory of error-correcting codes.

By definition, the *first batch of questions* of $\mathcal{S}$ is given by

> For each $i = 1, 2, \ldots, m$, let $\mathbf{D}_i = (D_{i,0}, \ldots, D_{i,k-1})$ denote the question "Which is the $i$th digit in the $k$-ary expansion of $x_{\text{Carole}}$?" Thus a number $y \in S$ belongs to $D_{i,j}$ iff the $i$th digit of its $k$-ary expansion $\vec{y} = y_1 \cdots y_m$ is equal to $j$.

Let $b_i \in \{0, 1, \ldots, k-1\}$ be Carole's answer to question $\mathbf{D}_i$. Let the string $\vec{b}$ of $k$-ary digits be defined by $\vec{b} = b_1 \cdots b_m$. Repeated application of (12) beginning with the initial state $\sigma = (S, \varnothing, \varnothing)$ shows that Paul's state of knowledge as an effect of Carole's answers is a triplet $\sigma^{\vec{b}} = (A_0, A_1, A_2)$, where

$A_0$ = the singleton containing the number whose $k$-ary expansion equals $\vec{b}$

$A_1 = \{y \in S \mid d_H(\vec{y}, \vec{b}) = 1\}$

$A_2 = \{y \in S \mid d_H(\vec{y}, \vec{b}) = 2\}$.

Thus the state $\sigma^{\vec{b}}$ has type $(1, m(k-1), \binom{m}{2}(k-1)^2)$. Moreover, repeated application of Proposition 5.1(i) (compare with [8]) yields $\text{ch}^{[k]}(\sigma^{\vec{b}}) = \text{ch}^{[k]}(k^m, 0, 0) - m$.

*The k-ary Critical Index $m_n^{[k]}$*

For each $m$-tuple $\vec{b} \in \{0, 1, \ldots, k-1\}^m$ given by Carole's answers, we shall construct a non-adaptive $k$-ary strategy with $\text{ch}^{[k]}(1, m(k-1), \binom{m}{2}(k-1)^2)$ questions, and show that the strategy is winning for the state $\sigma^{\vec{b}}$.

To this purpose, let us consider the values of $\text{ch}^{[k]}(1, m(k-1), \binom{m}{2}(k-1)^2)$ for $m \geq 1$.

DEFINITION 5.2. *Let $k \geq 2$ and $n \geq 3$ be arbitrary integers. The $k$-ary critical index $m_n^{[k]}$ is the largest integer $m \geq 0$ such that* $\text{ch}^{[k]}(1, m(k-1), \binom{m}{2}(k-1)^2) = n$.

LEMMA 5.3. *Let $n \geq 3$ be an arbitrary integer. Then for all $k \geq 2$ we have*

$$\left\lfloor \frac{\sqrt{2}k^{n/2}}{k-1} \right\rfloor - n - 1 \leq m_n^{[k]} \leq \left\lfloor \frac{\sqrt{2}k^{n/2}}{k-1} \right\rfloor - n + 1.$$

*Proof.* $m_n^{[k]}$ is the largest integer $m$ such that $w_n^{[k]}(1, m(k-1), \binom{m}{2}(k-1)^2) \le k^n$. A tedious but straightforward computation yields

$$m_n^{[k]} = \left\lfloor \frac{\sqrt{8k^n + k^2 - 6k + 1}}{2(k-1)} + \frac{k-3}{2(k-1)} - n \right\rfloor, \tag{14}$$

from which the desired conclusion follows immediately. ∎

The second batch of questions is obtainable from the following generalization of Lemma 4.4.

LEMMA 5.4. *Fix integers $a_0, a_1, a_2 \ge 0$, $k \ge 2$ and $n \ge \mathrm{ch}^{[k]}(a_0, a_1, a_2)$. Let $\sigma = (A_0, A_1, A_2)$ be a state of type $(a_0, a_1, a_2)$. Then there exists a non-adaptive winning $k$-ary strategy for $\sigma$ with $n$ questions* if and only if *for some integers $d_0 \ge 5$ and $d_1 \ge 3$ there exist an $(n, a_0, d_0)$ $k$-ary code $\mathscr{C}_0$ and an $(n, a_1, d_1)$ $k$-ary code $\mathscr{C}_1$, such that $\Delta(\mathscr{C}_0, \mathscr{C}_1) \ge 4$.*

*Proof.* The proof is a routine variant of the proof of Lemma 4.4. ∎

COROLLARY 5.5. *Fix arbitrary integers $k \ge 2$, $m \ge 1$, and assume the integer $n$ to satisfy the inequality $n \ge \mathrm{ch}^{[k]}(1, m(k-1), \binom{m}{2}(k-1)^2)$. Let $\sigma = (A_0, A_1, A_2)$ be a state of type $(1, m(k-1), \binom{m}{2}(k-1)^2)$. Then there exists a non-adaptive winning $k$-ary strategy for $\sigma$ with $n$ questions if and only if for some integer $d \ge 3$ there exists an $(n, m(k-1), d)$ $k$-ary code with minimum Hamming weight $\ge 4$.*

*Auxiliary Results*

LEMMA 5.6. *Let $k \ge 3$ and $n \ge 5$ be arbitrary integers. Then for some integer $M \ge m_n^{[k]}(k-1)$ there exists an $(n, M, 3)$ $k$-ary code $\mathscr{C}_{k,n}$ with $\mu(\mathscr{C}_{k,n}) \ge 4$.*

*Proof.* We shall first consider the cases

(i) $k = 3, n \ge 11$
(ii) $k = 4, 5, n \ge 9$
(iii) $6 \le k \le 8, n \ge 8$
(iv) $9 \le k \le 19, n \ge 7$
(v) $20 \le k \le 197, n \ge 6$
(vi) $k \ge 198, n \ge 5$.

In each of the above cases the desired result is a consequence of Lemmas 3.3 and 5.3, together with the inequalities

$$\frac{k^n - \sum_{i=0}^{3} \binom{n}{i}(k-1)^i}{\sum_{i=0}^{2} \binom{n}{i}(k-1)^i} \ge \sqrt{2}k^{\frac{n}{2}} \ge m_n^{[k]}(k-1).$$

We now consider the cases

(vii) $n = 5, \quad 4 \le k \le 197$
(viii) $n = 6, \quad 5 \le k \le 19$
(ix) $n = 7, \quad 5 \le k \le 8$
(x) $n = 8, \quad k = 5.$

*Case 1.* $k \ge n - 1$ is a prime power.

Then there exists an $(n, k^{n-2}, 3)$ $k$-ary code $\mathscr{D}_{k,n}$ [20]. Such a code belongs to the well known class of the MDS codes. In particular [13, Chap. 11, Theorem 6], $\mathscr{D}_{k,n}$ contains the codeword $\vec{0}$, has exactly $\binom{n}{3}(k-1)$ codewords with Hamming weight 3, and (because of $\delta(\mathscr{D}_{k,n}) = 3$) does not contain any non-zero codeword of Hamming weight $\le 2$. Upon defining $\mathscr{C}_{k,n} = \{\vec{x} \mid \vec{x} \in \mathscr{D}_{k,n}, w_H(\vec{x}) \ge 4\}$, by Lemma 5.3 we have

$$|\mathscr{C}_{k,n}| = k^{n-2} - \binom{n}{3}(k-1) - 1 \ge \sqrt{2}k^{n/2} - (n-1)(k-1) \ge m_n^{[k]}(k-1).$$

*Case 2.* $k > 5$ not a prime power and $(k, n) \ne (6, 7)$.

Then let $p_1$ be the largest prime power $< k$ and $p_2$ the smallest prime power $> k$. Let $d = p_2 - p_1$. Notice that, under our standing hypothesis, we have $p_1 \ge n - 1$. Let $\mathscr{C}_{k,n}$ be the code whose codewords are obtained from those of $\mathscr{D}_{p_1,n}$ (as defined in Case 1), replacing by the digit $p_1$ every occurrence of the zero digit. Trivially $\mathscr{C}_{k,n}$ is an $(n, p_1^{n-2}, 3)$ $k$-ary code and $\mu(\mathscr{C}_{k,n}) = n > 4$. Furthermore,

$$\begin{array}{ll} d \le 2 & \text{for } 5 < k < 13, \\ d \le 3 & \text{for } 13 < k < 17, \\ d \le 10 & \text{for } 17 < k < 197. \end{array} \tag{15}$$

Then the desired result follows from (15), together with Lemma 5.3 via the inequalities $|\mathscr{C}_{k,n}| = p_1^{n-2} \ge \sqrt{2}(p_1 + d - 1)^{n/2} - (n-1)(p_1 + d - 2) \ge \sqrt{2}k^{n/2} - (n-1)(k-1) \ge m_n^{[k]}(k-1)$ , which hold for any pair $(k, n)$ under consideration in the present case.

*Case 3.* $(k, n) \in \{(5, 7), (5, 8), (6, 7)\}$.

Again let $p_2$ be the smallest prime power $> k$. Thus, under our standing hypothesis, $p_2 \ge n - 1$. Let us define

$$\mathscr{C}_{k,n} = \{x_1 \cdots x_n \in \mathscr{D}_{p_2,n} \mid w_H(\vec{x}) \ge 4 \text{ and } x_i \le k - 1 \text{ for each } i = 1, \ldots, n\}.$$

In other words, $\mathscr{C}_{k,n}$ is the subcode of $\mathscr{D}_{p_2,n}$ (as defined in Case 1) whose codewords have Hamming weight $\ge 4$ and are defined on the $k$-ary alphabet $\{0, 1, \ldots, k-1\}$. We shall prove the inequality

$$|\mathscr{C}_{k,n}| \ge (k-1)m_n^{[k]}. \tag{16}$$

As a matter of fact, for every $(n, k^r, n-r+1)$ $k$-ary MDS code $\mathscr{C}$ and for any choice of distinct indices $i_1, \ldots, i_r \in \{1, 2, \ldots, n\}$ we have

$$\{x_{i_1} x_{i_2} \cdots x_{i_r} \mid x_1 \cdots x_n \in \mathscr{C}\} = \{0, 1, \ldots, k-1\}^r.$$

Therefore, for each $i = 1, 2, \ldots, n$ and $d \leq k$, there are exactly $dk^{r-1}$ codewords in $\mathscr{C}$ whose $i$th digit is in $\{k-d, k-d+1, \ldots, k-1\}$. Let $W \subseteq \mathscr{C}$ be the set of codewords whose initial segment of length $r-1$ contains at least one of the digits $\{k-d, k-d+1, \ldots, k-1\}$. It is not hard to verify that the number $\alpha_{d,r}^{[k]}$ of elements of $W$ is given by

$$\alpha_{d,r}^{[k]} = k(k^{r-1} - (k-d)^{r-1}).$$

Fix now $i \in \{r, r+1, \ldots, n\}$, and let $\beta_{d,r}^{[k]}$ be the number of codewords of $W$ whose $i$th digit is in $\{k-d, k-d+1, \ldots, k-1\}$. Then we have

$$\beta_{d,r}^{[k]} = d(k^{r-1} - (k-d)^{r-1}).$$

Let us now turn our attention to code $\mathscr{C}_{k,n}$. Recall that $\mathscr{D}_{p_2,n}$ is an $(n, p_2^{n-2}, 3)$ $p_2$-ary MDS code. Thus setting $d = p_2 - k$ and $r = n-2$, by the above consideration, upon deleting from $\mathscr{D}_{p_2,n}$ all codewords whose initial segment of length $n-3$ contains one of the digits $k, k+1, \ldots, p_2-1$, we obtain a new code $\mathscr{D}'_{p_2,n}$ having exactly $p_2^r - \alpha_{d,r}^{[p_2]}$ codewords. Furthermore, for each $i \in \{n-2, n-1, n\}$ there are at most $dp_2^{r-1} - \beta_{d,r}^{[p_2]}$ codewords in $\mathscr{D}'_{p_2,n}$ whose $i$th digit belongs to the set $\{k, k+1, \ldots, p_2-1\}$. Deleting from $\mathscr{D}'_{p_2,n}$ all these codewords we obtain a new code $\mathscr{D}''_{p_2,n}$ whose number of codewords is $\geq |\mathscr{D}'_{p_2,n}| - 3(dp_2^{r-1} - \beta_{d,r}^{[p_2]})$. Note that $\mathscr{D}''_{p_2,n}$ contains the codeword $\vec{0}$ and may well contain codewords with Hamming weight $= 3$. Since, as noted above, there are precisely $\binom{n}{3}(p_2-1)$ codewords of weight 3 in $\mathscr{D}_{p_2,n}$, we finally obtain

$$|\mathscr{C}_{k,n}| \geq \gamma_{k,n}^{[p_2]},$$

where

$$\gamma_{k,n}^{[p_2]} = p_2^r - \alpha_{d,r}^{[p_2]} - 3(dp_2^{r-1} - \beta_{d,r}^{[p_2]}) - \binom{n}{3}(k-1) - 1.$$

In conclusion, from the inequalities

$$\gamma_{5,7}^{[7]} = 414 > 4 \times 92 = 4 \times m_7^{[5]}$$

$$\gamma_{5,8}^{[7]} = 2788 > 4 \times 213 = 4 \times m_8^{[5]}$$

$$\gamma_{6,7}^{[7]} = 4973 > 5 \times 142 = 5 \times m_7^{[6]}$$

we have the desired result (16). Also Case 3 is settled.

We are now left with the nine cases

(xi) $k = 3, \quad n = 5, 6, 7, 8, 9, 10$

(xii) $k = 4, \quad n = 6, 7, 8.$

In the Appendix we display the desired codes $\mathscr{C}_{k,n}$ for each of these cases. Direct inspection shows that $|\mathscr{C}_{k,n}| = (k-1)m_n^{[k]}$. ∎

LEMMA 5.7. *For each integer $k \geq 2$ let $M_k$ be the largest integer $m$ such that there exists a $(4, mk, 3)$ $k$-ary code. Then*

(i) $M_k = k$ *for* $k = 3, 4, 5, 7, 8, 9, \ldots$

(ii) $M_k = k - 1$ *for* $k \in \{2, 6\}$.

*Proof.* By the well known Singleton bound (see [13] and references therein) each $(n, M, d)$ $k$-ary code satisfies the inequality $M \leq k^{n-d+1}$. Setting $n = 4$ and $d = 3$ we have the upper bound $M \leq k^2$, whence $M_k \leq k$.

To prove (i) we recall that finding a $(4, k^2, 3)$ $k$-ary code is equivalent to finding a pair of orthogonal Latin squares of order $k$ (see [13, Chap. 11] and references therein). It was proved by Bose et al. [4] that orthogonal Latin squares of order $n$ do exist for all integers $n \geq 2$, except for $n \in \{2, 6\}$. This settles (i).

In order to prove (ii), again by [4] we obtain $M_2 \leq 1$ and $M_6 \leq 5$. For the converse inequality, the $(4, 2, 3)$ binary code $\{0000, 1111\}$ and the $(4, 30, 3)$ 6-ary code

$$\{1000, 0110, 2220, 3330, 4440, 5550, 2101, 3011, 0321, 1231,$$

$$0202, 1312, 4022, 5132, 2542, 3452, 4303, 5213, 1423, 0533,$$

$$3143, 2053, 5404, 4514, 0044, 1154, 3505, 2415, 5345, 4255\}$$

are enough to show that $M_2 \geq 1$ and $M_6 \geq 5$. This settles (ii). ∎

COROLLARY 5.8. *For each integer $k \geq 3$ let $M_k$ be the largest integer $m$ such that there exists a $(4, m(k-1), 3)$ $k$-ary code $\mathscr{C}$ with $\mu(\mathscr{C}) = 4$. Then*

(i) $M_k = k - 1$ *for* $k = 4, 5, 6, 8, 9, \ldots$

(ii) $M_k = k - 2$ *for* $k \in \{3, 7\}$.

*Proof.* Let $k' = k - 1$. The existence of a $(4, M, 3)$ $k$-ary code $\mathscr{C}$, with $\mu(\mathscr{C}) = 4$, is equivalent to the existence of a $(4, M, 3)$ $k'$-ary code $\mathscr{C}'$. Indeed, no codeword in $\mathscr{C}$ can contain a digit equal to 0, because of $\mu(\mathscr{C}) = 4$, whence $\mathscr{C}$ is actually a $(4, M, 3)$ code defined over the $k'$-ary alphabet $\{1, 2, \ldots, k'\}$. Thus from $\mathscr{C}$ one immediately obtains a $k'$-ary code in the sense of Definition 3.1. Conversely, replacing by the digit $k$ each occurrence of the digit 0 in the codewords of a $(4, M, 3)$ $k'$-ary code $\mathscr{C}'$, we get a $(4, M, 3)$ $k$-ary code with $\mu(\mathscr{C}) = 4$. The desired conclusion now immediately follows by Lemma 5.7. ∎

THEOREM 5.9. *Fix integers* $k = 3, 4, \ldots$ *and* $m = 1, 2, \ldots$. *Let* $S = \{0, 1, \ldots, k^m - 1\}$ *and* $\lambda(k, m)$ *be the number of questions in a shortest* canonical *k-ary strategy to search for an unknown number* $x \in S$ *with two lies. Let the sets* $E_1, E_2, E_3$ *be defined by*

$$E_1 = \{(3, 2), (7, 6)\}$$

$$E_2 = \{(k, m) \in \mathbf{Z}^2 \mid k \geq 5 \text{ and } k \leq m \leq m_4^{[k]} = \left\lfloor \tfrac{\sqrt{8k^4+k^2-6k+1}}{2(k-1)} - \tfrac{7k-5}{2(k-1)} \right\rfloor\}$$

$$E_3 = \{(k, m) \in \mathbf{Z}^2 \mid k \geq 5 \text{ and } 1 \leq m \leq m_3^{[k]} = \left\lfloor \tfrac{\sqrt{8k^3+k^2-6k+1}}{2(k-1)} - \tfrac{5k-3}{2(k-1)} \right\rfloor\}.$$

*We then have*

(i) $E_2$ *coincides with the set of pairs* $(k, m) \in \mathbf{Z}^2$ *such that* $5 \leq k \leq m$ *and* $\mathrm{ch}(k^m, 0, 0) = m + 4$. $E_3$ *coincides with the set of pairs* $(k, m) \in \mathbf{Z}^2$ *such that* $\mathrm{ch}(k^m, 0, 0) = m + 3$. *Thus, in particular, for any fixed* $k$, *only finitely many integers* $m$ *are such that* $(k, m) \in E_2 \cup E_3$.

(ii) *In case* $(k, m) \notin E_1 \cup E_2 \cup E_3$ *then* $\lambda(k, m) = \mathrm{ch}^{[k]}(k^m, 0, 0)$.

(iii) *Otherwise, if* $(k, m) \in E_1 \cup E_2 \cup E_3$, *then* $\lambda(k, m) = \mathrm{ch}^{[k]}(k^m, 0, 0) + 1$.

*Proof.* After receiving Carole's answers to his first $m$ questions, Paul's state of knowledge $\sigma$ is of type $(1, m(k-1), \binom{m}{2}(k-1)^2)$. Furthermore we have $\mathrm{ch}^{[k]}(\sigma) = \mathrm{ch}^{[k]}(k^m, 0, 0) - m$. Therefore, for the proof of (ii) (resp., of (iii)), it suffices to show that the shortest non-adaptive winning $k$-ary strategy for $\sigma$ requires exactly $\mathrm{ch}^{[k]}(\sigma)$ (resp., $\mathrm{ch}^{[k]}(\sigma) + 1$) questions.

Writing $n$ as an abbreviation of $\mathrm{ch}^{[k]}(\sigma)$, direct inspection shows that $n \geq 3$. A tedious but straightforward computation using Lemma 5.3 settles (i).

(ii) Under the assumption $(k, m) \notin E_1 \cup E_2 \cup E_3$ we shall exhibit a $k$-ary non-adaptive winning strategy for $\sigma$ using $n$ questions. We argue by cases:

*Case 1.* $n \geq 5$.

Then by Lemma 5.6 there exists an $(n, M, 3)$ $k$-ary code $C_{k,n}$ such that $\mu(C_{k,n}) \geq 4$ and $M \geq m_n^{[k]}(k-1)$. By Definition 5.2, $M \geq m_n^{[k]}(k-1) \geq m(k-1)$. The desired conclusion now follows from Corollary 5.5 by picking a subcode $\mathscr{D}_{k,m} \subseteq \mathscr{C}_{k,n}$ with $|\mathscr{D}_{k,m}| = m(k-1)$.

*Case 2.* $n = 4$.

By direct inspection, for each $k \in \{3, 4\}$, we have $m_4^{[k]} = k - 1$. By our standing hypothesis $(k, m) \notin E_2$ we easily obtain $m \leq (k-1)$.

Let $k \notin \{3, 7\}$. By Corollary 5.8 there exists a $(4, (k-1)^2, 3)$ $k$-ary code $\mathscr{C}$ with $\mu(\mathscr{C}) \geq 4$. Picking any subcode $\mathscr{D} \subseteq \mathscr{C}$, with $|\mathscr{D}| = m(k-1)$, and using Corollary 5.5 we have the desired strategy.

If $k \in \{3, 7\}$ then from $(k, m) \notin E_1$ we get $m \leq k-2$. Again by Corollary 5.8 there exists a $(4, (k-2)(k-1), 3)$ $k$-ary code $\mathscr{C}$ with $\mu(\mathscr{C}) \geq 4$. Picking now any subcode $\mathscr{D} \subseteq \mathscr{C}$, with $|\mathscr{D}| = m(k-1)$ and using Corollary 5.5 we get the desired strategy.

*Case 3.* $n = 3$.

This case would imply $(k, m) \in E_3$, which is impossible.

The proof of (ii) is complete.

(iii) Under the assumption $(k, m) \in E_1 \cup E_2 \cup E_3$, let $\xi(\sigma)$ denote the length of the shortest winning $k$-ary strategy for $\sigma$. We prove that $\xi = n+1$. We shall again argue by cases as follows:

*Case* I. $(k, m) \in E_1 \cup E_2$.

Then $n = 4$. By Corollaries 5.8 and 5.5 no winning $k$-ary strategy can exist for $\sigma$ with four questions, whence $\xi(\sigma) \geq 5$. On the other hand, $m \leq m_4^{[k]} < m_5^{[k]}$. Thus by Lemma 5.6 there exists a $(5, M, 3)$ $k$-ary code $\mathscr{C}$ such that $\mu(\mathscr{C}) \geq 4$ and $M \geq m_5^{[k]}(k-1) > m(k-1)$. Picking now a subcode $\mathscr{D} \subseteq \mathscr{C}$, with $|\mathscr{D}| = m(k-1)$, and using Corollary 5.5 we get $\xi(\sigma) = 5 = \mathrm{ch}^{[k]}(\sigma) + 1$, as required.

*Case* II. $(k, m) \in E_3$.

We then have $n = 3$, $k \geq 5$, and

$$m \leq \left\lfloor \frac{\sqrt{8k^3 + k^2 - 6k + 1}}{2(k-1)} - \frac{5k-3}{2(k-1)} \right\rfloor < (k-3).$$

Trivially, no $(3, m(k-1), 3)$ $k$-ary code $\mathscr{C}$ with $\mu(\mathscr{C}) = 4$ can exist; hence by Corollary 5.5 we have $\xi(\sigma) \geq 4 = \mathrm{ch}^{[k]}(\sigma) + 1$. Moreover, by Corollary 5.8, for some $M \geq (k-1)(k-2) > m(k-1)$ there exists a $(4, M, 3)$ $k$-ary code $\mathscr{C}$ such that $\mu(\mathscr{C}) = 4$. Picking any subcode $\mathscr{D} \subseteq \mathscr{C}$ such that $|\mathscr{D}| = m(k-1)$ and using Corollary 5.5, we obtain a non-adaptive winning strategy for $\sigma$ with $\mathrm{ch}^{[k]}(\sigma) + 1$ questions. ∎

PROPOSITION 5.10. *Adopt the above notation. For each pair* $(k, m) \in E_2 \cup E_3$ *and state* $\sigma$ *of type* $(k^m, 0, 0)$, *there is no perfect strategy for* $\sigma$ *with two lies—even in the fully adaptive model.*

*Proof.* Let $\sigma = (A_0, \varnothing, \varnothing)$ be a state of type $(k^m, 0, 0)$. Let $\tau = (\varnothing, A_0, \varnothing)$. Let $\mathbf{T}_1, \ldots, \mathbf{T}_m$ be the first $m$ questions of a winning strategy $\mathscr{S}$ for $\sigma$. We shall prove that the number $t$ of questions of $\mathscr{S}$ necessarily satisfies the inequality $t > \mathrm{ch}^{[k]}(k^m, 0, 0)$, whence $\mathscr{S}$ cannot be perfect. For every sequence $\vec{r} = r_1 \cdots r_m$ of Carole's answers ($r_j \in \{0, \ldots, k-1\}$) to $\mathbf{T}_1, \ldots, \mathbf{T}_m$ let $\sigma^{\vec{r}}$ denote the state resulting from these answers. As an effect of these answers, also $\tau$ is transformed into a new state $\tau^{\vec{r}}$. Repeated application of Proposition 5.1(i) shows that, among Carole's answering strategies

$\vec{r} = r_1 \cdots r_m$ to the above questions, at least one, say $\vec{z} = z_1 \cdots z_m$, satisfies the inequality

$$w_{t-m}^{[k]}(0, a, b) \geq \frac{w_t^{[k]}(0, k^m, 0)}{k^m} = w_{t-m}^{[k]}(0, 1, m(k-1)), \tag{17}$$

where $(a, b, c)$ is the type of $\sigma^{\vec{z}}$, and hence, $(0, a, b)$ is the type of $\tau^{\vec{z}}$.

It follows that, for every $t > m$,

$$w_{t-m}^{[k]}(0, a, b) > k \qquad \text{whenever } m \geq 1, \tag{18}$$

and, moreover,

$$w_{t-m}^{[k]}(0, a, b) > k^2 \qquad \text{whenever } m \geq k. \tag{19}$$

Therefore, recalling (18), for all $m \geq 1$ we can write $\mathrm{ch}^{[k]}(0, a, b) \geq 2$. By [8, translation bound] the smallest winning strategy for $\sigma^{\vec{z}}$ requires $\geq 2 + \mathrm{ch}^{[k]}(0, a, b) \geq 4$ questions. Therefore, $t - m \geq 4$, and, recalling Theorem 5.9(i), we see that $\mathscr{S}$ is not perfect in case $(k, m) \in E_3$.

Now turning attention to the case $(k, m) \in E_2$, in the light of (19) for all $m \geq k$ we can write $\mathrm{ch}^{[k]}(0, a, b) \geq \mathrm{ch}^{[k]}(0, 1, m(k-1)) \geq 3$. Again by [8, translation bound], the smallest winning strategy for $\sigma^{\vec{z}}$ requires at least $2 + \mathrm{ch}^{[k]}(0, a, b) \geq 5$ questions. Since the strategy is assumed to be winning then $t - m \geq 5$, whence $t \geq m + 5$. Again, $\mathscr{S}$ is not perfect, for all pairs $(k, m) \in E_2$. ∎

*Further Preparatory Material*

The following lemma provides *perfect* $k$-ary strategies with minimum adaptiveness for each case $(k, m) \in E_1$, defined in Theorem 5.9. In view of Theorem 5.9(iii), our perfect strategies here are *non-canonical*: indeed, every strategy considered in this section uses a first batch of $m + 1$ (rather than $m$) non-adaptive questions. Paul's state resulting from Carole's answers to these questions carries more information than Carole's answers to the first batch of questions in any canonical strategy. A carefully designed second batch of $\mathrm{ch}^{[k]}(k^m, 0, 0) - m - 1$ questions allows Paul to infallibly guess Carole's number.

LEMMA 5.11. *For each* $k = 2, 3, \ldots$ *there exists a* perfect *$k$-ary strategy to guess a number* $x \in S = \{0, \ldots, k^{k-1}\}$ *with up to two lies in the answers, using a first batch of* $k$ *non-adaptive questions and then, only depending on the answers to these questions, a second batch of* $\mathrm{ch}^{[k]}(k^{k-1}, 0, 0) - k$ *non-adaptive questions.*

*Proof.* Again, let $x_{\text{Carole}}$ denote Carole's secret number. Direct computation shows that $\mathrm{ch}^{[k]}(k^{k-1}, 0, 0) = k + 3$. Our first batch of $k$ non-adaptive questions, denoted $[\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_k]$, is defined as follows:

For each $i = 1, \ldots, k-1$, question $\mathbf{Q}_i$ asks "What is the $i$th digit in the $k$-ary expansion of $x_{\text{Carole}}$?" Finally, question $\mathbf{Q}_k$ asks "What is the sum (modulo $k$) of the digits in the $k$-ary expansion of $x_{\text{Carole}}$?"

*Claim.* Only depending on Carole's reply to questions $Q_1, \ldots, Q_k$, the resulting state is of either type $(1, 0, \binom{k}{2}(k-1))$ or $(0, k, \binom{k}{2}(k-2))$.

As a matter of fact, let the map $x \in S \mapsto \vec{x} = x_1 x_2 \cdots x_k \in \{0, 1, \ldots, k-1\}^k$ be defined by

$$x_i = \text{the } i\text{th digit in the } k\text{-ary expansion of } x \qquad (i = 1, \ldots, k-1)$$
$$x_k \equiv x_1 + \cdots + x_{k-1} \bmod k.$$

Let $\vec{a} = a_1 a_2 \cdots a_k$ be the $k$-tuple of Carole's answers, $a_i \in \{0, \ldots, k-1\}$. There exists exactly one $x \in S$ such that $x_1 \cdots x_{k-1} = a_1 \cdots a_{k-1}$.

*Case 1.* $x_k = a_k$.

Then $\vec{x} = \vec{a}$ and Paul's state of knowledge after Carole's first $k$ answers is of type $(1, r, s)$ for some $r, s \geq 0$. We shall prove

$$r = 0 \qquad \text{and} \qquad s = \binom{k}{2}(k-1). \tag{20}$$

Let us suppose $r > 0$ (absurdum hypothesis), and let $h \in S$ be such that $d_H(\vec{h}, \vec{x}) = 1$. Since by definition, $d_H(h_1 \cdots h_{k-1}, x_1 \cdots x_{k-1}) \geq 1$ it follows that $d_H(h_1 \cdots h_{k-1}, x_1 \cdots x_{k-1}) = 1$. Let $i \in \{1, \ldots, k-1\}$ be such that $x_i \neq h_i$. We then have $h_i \equiv x_i + d_i \bmod k$ for a uniquely determined $d_i \in \{1, \ldots, k-1\}$. Therefore, modulo $k$, we can write the congruences

$$h_k \equiv \sum_{j=1}^{k-1} h_j \equiv \left(\sum_{j=1}^{k-1} x_j\right) + d_i \not\equiv \left(\sum_{j=1}^{k-1} x_j\right) \equiv x_k, \tag{21}$$

thus obtaining the contradiction $d_H(\vec{x}, \vec{h}) = 2$. This shows $r = 0$.

To end the proof of (20), let $s'$ be the number of elements $y \in S$ such that $d_H(x_1 \cdots x_{k-1}, y_1 \cdots y_{k-1}) = 1$ and $y_k \neq x_k$; further let $s''$ be the number of $y \in S$ such that $d_H(x_1 \cdots x_{k-1}, y_1 \cdots y_{k-1}) = 2$ and $y_k = x_k$. Then $s = s' + s''$. To compute the value of $s'$, suppose $y \in S$ to satisfy $d_H(x_1 \cdots x_{k-1}, y_1 \cdots y_{k-1}) = 1$ and $y_k \neq x_k$. Let $i \in \{1, \ldots, k-1\}$ be the only index such that $y_i \neq x_i$ and write $y_i \equiv x_i + d_i \bmod k$ for a uniquely determined $d_i \in \{1, \ldots, k-1\}$. An easy modification of (21) shows that the number of such elements $y$ is equal to the number of possible choices of index $i = 1, \ldots, k-1$ multiplied by the number of choices of $d_i = 1, \ldots, k-1$. Therefore, $s' = (k-1)^2$.

In order to compute $s''$ let $y \in S$ be such that $d_H(x_1 \cdots x_{k-1}, y_1 \cdots y_{k-1}) = 2$ and $y_k = x_k$. There exist exactly two indices $1 \le i < j \le k-1$ such that $x_i \neq y_i$ and $x_j \neq y_j$. Write $y_i \equiv x_i + d_i \bmod k$ and $y_j \equiv x_j + d_j \bmod j$ for suitable $d_i, d_j \in \{1, \ldots, k-1\}$. We have the following congruences, modulo $k$:

$$\sum_{j=1}^{k-1} x_j \equiv x_k \equiv y_k \equiv \sum_{j=1}^{k-1} y_j \equiv \left(\sum_{j=1}^{k-1} x_j\right) + d_i + d_j. \tag{22}$$

Thus, $d_i + d_j \equiv 0 \bmod k$, whence $d_i \equiv k - d_j$. Since every choice of $d_i$ uniquely determines $d_j$ there are precisely $k-1$ ways to choose $d_i, d_j$ and $\binom{k-1}{2}$ ways to choose $i, j$. We have shown that $s'' = \binom{k-1}{2}(k-1)$, as required.

Summing up, $s = \left[\binom{k-1}{2} + (k-1)\right](k-1) = \binom{k}{2}(k-1)$, and Eq. (20) holds.

*Case 2.* $a_k \neq x_k$.

Then Paul's state after Carole's answers is of type $(0, t, u)$, with $t \ge 1$ (because $d_H(\vec{x}, \vec{a}) = 1$) and $u \ge 0$. We shall first prove

$$t = k. \tag{23}$$

To this purpose, let us write $a_k \equiv x_k + d_k \bmod k$, for a uniquely determined $d_k \in \{1, \ldots, k-1\}$. For every $y \in S, y \neq x$ we have $d_H(a_1 \cdots a_{k-1}, y_1 \cdots y_{k-1}) \ge 1$. For the identity $d_H(\vec{a}, \vec{y}) = 1$ to hold, we must have $d_H(a_1 \cdots a_{k-1}, y_1 \cdots y_{k-1}) = 1$ and $\sum_{j=1}^{k-1} y_j \equiv a_k \bmod k$. Let $i \in \{1, \ldots, k-1\}$ be uniquely determined by the condition $y_i \neq x_i = a_i$. Then $y_i \equiv x_i + d_i \bmod k$ for a unique $d_i \in \{1, \ldots, k-1\}$. Thus, modulo $k$, we can write the congruences

$$\left(\sum_{j=1}^{k-1} x_j\right) + d_i \equiv \sum_{j=1}^{k-1} y_j \equiv a_k \equiv x_k + d_k \equiv \left(\sum_{j=1}^{k-1} x_j\right) + d_k,$$

showing that $d_k = d_i$. It follows that the number of elements $y \in S$ whose $k$-ary coding has distance 1 from $\vec{a}$ is equal to the number $k-1$ of possible choices of the index $i$, plus the single contribution given by $x$ itself. This shows (23).

We shall now prove

$$u = \binom{k}{2}(k-2). \tag{24}$$

Let $U'$ be the set of $y \in S$ such that $d_H(y_1 \cdots y_{k-1}, x_1 \cdots x_{k-1}) = 1$ and $y_k \neq a_k$; further, let $U''$ be the set of $y \in S$ such that $d_H(y_1 \cdots y_{k-1}, x_1 \cdots x_{k-1}) = 2$ and $y_k \equiv a_k \equiv x_k + d_k \bmod k$. Let $u'$ and $u''$ be the number of elements of $U'$ and $U''$, respectively.

Assume $y \in U'$. There is $i \in \{1, \ldots, k-1\}$ such that $y_i \equiv x_i + d_i \bmod k$ for some $d_i \in \{1, \ldots, k-1\}$. Since $y_k \neq a_k$, we can write the following congruences modulo $k$:

$$\left(\sum_{j=1}^{k-1} x_j\right) + d_i \equiv \sum_{j=1}^{k-1} y_j \equiv y_k \not\equiv x_k + d_k \equiv \left(\sum_{j=1}^{k-1} x_j\right) + d_k.$$

Thus, $d_i \neq d_k$. Since $d_i \in \{1, \ldots, k-1\} \setminus \{d_k\}$ we see that $U'$ has as many elements as the number $k-1$ of possible choices for $i$ multiplied by the number $k-2$ of possible choices for $d_i$. Therefore, $u' = (k-1)(k-2)$.

Assume now $y \in U''$. Then there exist exactly two indices $1 \leq i < j \leq k-1$, together with elements $d_i, d_j \in \{1, \ldots, k-1\}$, such that $y_i \equiv x_i + d_i \bmod k$ and $y_j \equiv x_j + d_j \bmod k$. We can write the following congruences modulo $k$:

$$\left(\sum_{j=1}^{k-1} x_j\right) + d_i + d_j \equiv \sum_{j=1}^{k-1} y_j \equiv y_k \equiv x_k + d_k \equiv \left(\sum_{j=1}^{k-1} x_j\right) + d_k.$$

It follows that $d_i + d_j \equiv d_k \bmod k$. Then any element of $U''$ is uniquely determined by the choice of the pair of indices $i, j$ together with the choice of $d_i \in \{1, \ldots, k-1\} \setminus \{d_k\}$. Specifically, we must have $d_i \neq d_k$ because we must ensure $d_j \neq 0$ and $d_i + d_j \equiv d_k \bmod k$. Thus there are exactly $u'' = \binom{k-1}{2}(k-2)$ elements $y \in U''$. Summing up we have $u = u' + u'' = [\binom{k-1}{2} + (k-1)](k-2) = \binom{k}{2}(k-2)$, which establishes (24) and also concludes the proof of the claim.

To conclude the proof, assume the state $\sigma$ resulting from Carole's answers to $[\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_k]$ is of type $(1, 0, \binom{k}{2}(k-1))$. Direct computation yields $\mathrm{ch}^{[k]}(\sigma) = 3$. In view of Lemma 5.4, the trivial $(3, 1, 5)$ $k$-ary code $\mathscr{C}_0 = \{000\}$, along with the empty code $\mathscr{C}_1 = \varnothing$, provides the desired result.

For the other case, assume the state $\sigma$ resulting from Carole's answers to $[\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_k]$ has type $(0, k, \binom{k}{2}(k-2)$. Again, $\mathrm{ch}^{[k]}(\sigma) = 3$. One more application of Lemma 5.4 using the codes $\mathscr{C}_0 = \varnothing$ and $\mathscr{C}_1 = \{www \mid w = 0, 1, \ldots, k-1\}$ yields a non-adaptive $k$-ary winning strategy for $\sigma$ with three questions. The proof is complete. ∎

## 6. MAIN RESULTS AND FINAL REMARKS

By Berlekamp's bound (Proposition 5.1), every winning $k$-ary strategy $\mathscr{S}$ to guess a number $x \in \{0, \ldots, k^m - 1\}$ with up to two lies in the answers must necessarily use *at least* $\mathrm{ch}^{[k]}(k^m, 0, 0)$ questions. As the reader will recall, when $\mathscr{S}$ uses *exactly* $\mathrm{ch}^{[k]}(k^m, 0, 0)$ questions, $\mathscr{S}$ is said to be *perfect*. We say that $\mathscr{S}$ is *quasi perfect* if it uses $1 + \mathrm{ch}^{[k]}(k^m, 0, 0)$ questions.

The following result sums up what is known about the existence and non-existence of perfect and quasi perfect searching strategies *with minimum adaptiveness* and up to two faulty tests.

THEOREM 6.1. *Let the sets* $S_1, S_2, S_3$ *be defined by*

(i) $S_1 = \{(2,2), (2,4), (3,2), (7,6)\}$

(ii) $S_2 = \{(k,m) \in \mathbf{Z}^2 \mid 5 \le k \le m \text{ and } \mathrm{ch}(k^m, 0, 0) = m + 4\}$

(iii) $S_3 = \{(k,m) \in \mathbf{Z}^2 \mid \mathrm{ch}(k^m, 0, 0) = m + 3\}$.

*We then have*

(A) *For any fixed* $k = 2, 3, 4, \ldots$, *there are only finitely many integers* $m$ *such that* $(k,m) \in S_1 \cup S_2 \cup S_3$.

(B) *For all pairs of integers* $(k,m)$, $k \ge 2, m \ge 1$, *other than those listed in* (i)–(iii), *there is a* perfect *and* canonical $k$*-ary strategy* $\mathcal{S}$ *to guess a number* $x \in \{0, \ldots, k^m - 1\}$ *with up to two lies in the answers.*[6]

(C) *A* non-canonical perfect *strategy* $\mathcal{I}$ *also exists for the pair* $(2,4)$, *as well as for the pairs* $(3,2)$ *and* $(7,6)$ (*more generally, for each pair* $(k, k-1)$, *where* $k = 2, 3, \ldots$). *Strategy* $\mathcal{I}$ *uses a first batch of* $1 + m$ *non-adaptive questions and then, only depending on the answers to these questions, a second batch of* $\mathrm{ch}^{[k]}(k^m, 0, 0) - m - 1$ *non-adaptive questions.*

(D) *The pair* $(2,2)$, *as well as all pairs listed in* (ii) *and* (iii), *has a* quasi perfect *strategy, using a first batch of* $m$ *non-adaptive questions and then, only depending on the answers to these questions, a second batch of* $1 + \mathrm{ch}^{[k]}(k^m, 0, 0) - m$ *non-adaptive questions. None of these pairs has a perfect strategy, even in the fully adaptive game.*

*Proof.* (A) By Theorem 5.9(i) together with Eq. (14) we have

$$S_2 = \{(k,m) \in \mathbf{Z}^2 \mid k \ge 5 \text{ and } m = k, k+1, \ldots, \left\lfloor \frac{\sqrt{8k^4 + k^2 - 6k + 1}}{2(k-1)} - \frac{7k-5}{2(k-1)} \right\rfloor\}$$

and

$$S_3 = \{(k,m) \in \mathbf{Z}^2 \mid k \ge 5 \text{ and } m = 1, 2, \ldots, \left\lfloor \frac{\sqrt{8k^3 + k^2 - 6k + 1}}{2(k-1)} - \frac{5k-3}{2(k-1)} \right\rfloor\}.$$

This immediately yields the desired conclusion.

[6]Thus, by definition, $\mathcal{S}$ uses a first batch of $m$ non-adaptive questions asking for the $k$-ary digits of $x$ and then, only depending on the answers to these questions, a second batch of $\mathrm{ch}^{[k]}(k^m, 0, 0) - m$ non-adaptive questions.

(B) This follows from Theorems 5.9(ii) and 4.10.

(C) This follows from Lemmas 4.11 and 5.11.

(D) The first statement follows from Theorem 4.10, together with Theorem 5.9(iii). For the second statement, all pairs listed in (ii) and (iii) are taken care of by Proposition 5.10. Concerning the pair $(2, 2)$, we first note that $\mathrm{ch}^{[2]}(4, 0, 0) = 7$. Further, $\mathrm{ch}^{[2]}(0, 4, 0) = 5$. By [3, translation bound], the shortest winning strategy for a state of type $(4, 0, 0)$ requires at least $3 + \mathrm{ch}^{[2]}(0, 4, 0) = 8$ questions, whence no winning strategy can exist with less than $\mathrm{ch}^{[2]}(4, 0, 0) + 1$ questions. (Compare with the argument in [9, pp. 75–76].) ∎

*Remarks.* (1) For each $(k, m)$ the above result provides an *optimal* two-fault tolerant search strategy $\mathcal{S}$ for Paul to find an unknown number $x \in S = \{0, 1, \ldots, k^m - 1\}$. With respect to fully non-adaptive strategies considered in two-error correcting theory (where optimality results are rather the exception), $\mathcal{S}$ has the property that Paul is allowed, once and only once, to adapt his searching strategy. Thus Theorem 6.1 significantly strengthens the optimality results proved in [8, 9] for the fully adaptive model.

(2) As is well known, the Ulam–Rényi game has an equivalent *cooperative* formulation where, to fix ideas, Carole is identified with a satellite and Paul is the receiver; answers are now $k$-ary digits transmitted by the satellite via a noisy channel. Noise has the same effect as lies/errors. For this cooperative model, the results of our paper show that the minimum amount of redundancy which, by Berlekamp's bound, is *necessary* for two-error correction of an $m$-tuple of $k$-ary digits, turns out to be *sufficient*—on the condition that the receiver is allowed to send just one feedback message to the satellite. This is achieved via the following protocol:

(i) the original $m$-tuple $\vec{x}$ is sent by the satellite and is received as $\vec{x}'$;

(ii) the receiver feeds $\vec{x}'$ back to the satellite via a noiseless feedback channel as in [3];

(iii) only depending on $\vec{x}'$, the satellite sends a final tip $\vec{r}$ of $\mathrm{ch}^{[k]}(k^m, 0, 0) - m$ many $k$-ary digits, which are received as $\vec{r}'$, in such a way that

(iv) from $\vec{x}'\vec{r}'$ the receiver is able to recover the original $m$-tuple $\vec{x}$ (as well as $\vec{r}$), even if distortion has corrupted one or two of the digits of $\vec{x}\vec{r}$ (causing $\vec{x}'\vec{r}'$ to be received instead of $\vec{x}\vec{r}$).

(3) In this paper, perfect two-error-correcting minimum feedback strategies are effectively computed, for all $m \neq 2$, by an inductive procedure based on several error-correcting codes actually existing in the literature.

This is not the same as giving nonconstructive existence proofs for suitably large $m$. For instance, our analysis has shown that, for $m = 4$, perfect two-error-correction can only be achieved if the receiver sends the feedback acknowledgment after receiving $m + 1$ (rather than $m$) digits. This increases by one digit the feedback size to be transmitted via the noiseless channel during step (ii) in the above protocol.

## REFERENCES


1. M. Aigner, Searching with lies, *J. Combin. Theory Ser. A* **74** (1995), 43–56.
2. R. S. Borgstrom and S. Rao Kosaraju, Comparison-based search in the presence of errors, *in* "Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, San Diego, California, 16–18 May 1993," pp. 130–136.
3. E. R. Berlekamp, Block coding for the binary symmetric channel with noiseless, delayless feedback, *in* "Error-Correcting Codes" (H.B. Mann, Ed.), pp. 61–88, Wiley, New York, 1968.
4. R.C. Bose, S. S. Shrikhande, and E. T. Parker, Further results in the construction of mutually orthogonal Latin squares and the falsity of a conjecture of Euler, *Canad. J. Math*. **12** (1960), 189–203.
5. A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith, A new table of constant weight codes, *IEEE Trans. Inform. Theory* **36** (1990), 1334–1380.
6. F. Cicalese, $Q$-ary searching with lies, *in* "Proc. of the Sixth Italian Conf. on Theoretical Computer Science" (P. Degano, U. Vaccaro, and G. Pirillo, Eds.), pp. 228–240, World Scientific, Singapore, 1998.
7. F. Cicalese and D. Mundici, Optimal binary search with two unreliable tests and minimum adaptiveness, *in* "Proc. of European Symposium on Algorithms (ESA' 99)" (J. Nesetril, Ed.), Lectures Notes in Computer Science, Vol. **1643**, pp. 257–266, Springer-Verlag, Berlin/New York, 1999.
8. F. Cicalese and U. Vaccaro, Optimal strategies against a liar, *Theoret. Comput. Sci*. **230** (2000), 167–193.
9. J. Czyzowicz, D. Mundici, and A. Pelc, Ulam's searching game with lies, *J. Combin. Theory Ser. A* **52** (1989), 62–76.
10. R. Hill, Searching with lies, *in* "Surveys in Combinatorics" (P. Rowlinson, Ed.), pp. 41–70, Cambridge Univ. Press, Cambridge, UK, 1995.
11. R. Hill, J. Karim, and E. R. Berlekamp, The solution of a problem of Ulam on searching with lies, *in* "IEEE ISIT 1998, Cambridge, MA," p. 244.
12. E. Knill, Lower bounds for identifying subset members with subset queries, *in* "Proc. of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms," pp. 369–377, 1995.
13. F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error-Correcting Codes," North-Holland, Amsterdam, 1977.
14. D. Mundici and A. Trombetta, Optimal comparison strategies in Ulam's searching game with two errors, *Theoret. Comp. Sci*. **182** (1997), 217–232.
15. A. Negro and M. Sereno, Ulam's searching game with three lies, *Adv. Appl. Math*. **13** (1992), 404–428.
16. A. Pelc, Solution of Ulam's problem on searching with a lie, *J. Combin. Theory Ser. A* **44** (1987), 129–142.
17. A. Pelc, Searching with permanently faulty tests, *Ars Combin*. **38** (1994), 65–76.
18. A. Rényi, "Napló az információelméletről," Gondolat, Budapest, 1976 (English translation: "A Diary on Information Theory," Wiley, New York, 1984).

19. R. L. Rivest, A. R. Meyer, D. J. Kleitman, K. Winklmann, and J. Spencer, Coping with errors in binary search procedures, *J. Comput. System Sci*. **20** (1980), 396–404.
20. R. C. Singleton, Maximum distance $q$-nary codes, *IEEE Trans. Inform. Theory* **10** (1964), 116–118.
21. J. Spencer, Ulam's searching game with a fixed number of lies, *Theoret. Comput. Sci*. **95** (1992), 307–321.
22. A. Tietäväinen, On the nonexistence of perfect codes over finite fields, *SIAM J. Appl. Math*. **24** (1973), 88–96.
23. S. M. Ulam, "Adventures of a Mathematician," Scribner's, New York, 1976.

## APPENDIX: THE LAST NINE CODES OF LEMMA 5.6

We use the same notation as in the proof of Lemma 3.2, together with the abbreviation $\vec{0}^j = \underbrace{0 \cdots 0}_{j \text{ times}}$. For each $n \in \{5, 6, 7, 8, 9, 10\}$ the code $\mathscr{C}_{3,n}$ is given by

$$\mathscr{C}_{3,n} = \bigcup_{i=5}^{n} \mathscr{D}_{3,i} \otimes \vec{0}^{n-i},$$

where $\mathscr{D}_{3,5}, \ldots, \mathscr{D}_{3,10}$ are as follows in Table 1.

For each $n \in \{6, 7, 8\}$ the code $\mathscr{C}_{4,n}$ is given by

$$\mathscr{C}_{4,n} = \bigcup_{i=6}^{n} \mathscr{D}_{4,i} \otimes \vec{0}^{n-i},$$

where $\mathscr{D}_{4,6}, \ldots, \mathscr{D}_{4,8}$ are as follows in Table 2.

TABLE 1
The codes $\mathscr{C}_{3,n}$, for $n = 5, 6, 7, 8, 9, 10$.

| | | | | | |
|---|---|---|---|---|---|
| **1.** 01111 | **2.** 02222 | **3.** 10211 | **4.** 20122 | **5.** 21021 | **6.** 12012 |
| **7.** 22201 | **8.** 11102 | **9.** 22110 | **10.** 11220 | **11.** 12121 | **12.** 21212 |
| | | **The subcode $\mathscr{D}_{3,5}$.** | | | |
| **13.** 211001 | **14.** 122001 | **15.** 202101 | **16.** 220201 | **17.** 101201 | **18.** 021011 |
| **19.** 002211 | **20.** 012021 | **21.** 001121 | **22.** 110221 | **23.** 110102 | **24.** 022102 |
| **25.** 011202 | **26.** 120012 | | | | |
| | | **The subcode $\mathscr{D}_{3,6}$.** | | | |
| **27.** 2010120 | **28.** 2100220 | **29.** 1020220 | **30.** 1210001 | **31.** 2120001 | **32.** 2001101 |
| **33.** 0221101 | **34.** 0012101 | **35.** 2200201 | **36.** 0101201 | **37.** 1011201 | **38.** 2101011 |
| **39.** 0211011 | **40.** 0122011 | **41.** 1020111 | **42.** 1201111 | **43.** 1112111 | **44.** 2222111 |
| **45.** 2010211 | **46.** 1121211 | **47.** 0002211 | **48.** 2011021 | **49.** 1002021 | **50.** 0100121 |
| **51.** 2121121 | **52.** 0210221 | | | | |
| | | **The subcode $\mathscr{D}_{3,7}$.** | | | |
| **53.** 00212210 | **54.** 21122210 | **55.** 12222210 | **56.** 11110020 | **57.** 12020020 | **58.** 20220020 |
| **59.** 11001020 | **60.** 01221020 | **61.** 12202020 | **62.** 02112020 | **63.** 10010120 | **64.** 20001120 |
| **65.** 01011120 | **66.** 22111120 | **67.** 02002120 | **68.** 01122120 | **69.** 10222120 | **70.** 22000220 |
| **71.** 10100220 | **72.** 01200220 | **73.** 00111220 | **74.** 12211220 | **75.** 02021220 | **76.** 20012220 |
| **77.** 21010001 | **78.** 12210001 | **79.** 02120001 | **80.** 21101001 | **81.** 02011001 | **82.** 10021001 |
| **83.** 20202001 | **84.** 01022001 | **85.** 01210101 | **86.** 22001101 | **87.** 11201101 | **88.** 20111101 |
| **89.** 01121101 | **90.** 10002101 | **91.** 11112101 | **92.** 22212101 | **93.** 11100201 | **94.** 22200201 |
| **95.** 00110201 | **96.** 20020201 | **97.** 00201201 | **98.** 12111201 | | |
| | | **The subcode $\mathscr{D}_{3,8}$.** | | | |
| **99.** 212212010 | **100.** 020022010 | **101.** 101222010 | **102.** 201000110 | **103.** 022000110 | **104.** 110200110 |
| **105.** 120010110 | **106.** 221110110 | **107.** 112110110 | **108.** 202210110 | **109.** 011020110 | **110.** 121220110 |
| **111.** 102101110 | **112.** 020201110 | **113.** 211201110 | **114.** 210021110 | **115.** 002021110 | **116.** 010102110 |
| **117.** 002202110 | **118.** 200012110 | **119.** 021212110 | **120.** 112022110 | **121.** 100122110 | **122.** 220222110 |
| **123.** 102000210 | **124.** 001010210 | **125.** 220210210 | **126.** 111210210 | **127.** 221020210 | **128.** 000120210 |
| **129.** 212220210 | **130.** 010001210 | **131.** 121001210 | **132.** 220101210 | **133.** 001201210 | **134.** 112201210 |
| **135.** 022011210 | **136.** 212111210 | **137.** 200221210 | **138.** 211102210 | **139.** 110112210 | **140.** 201212210 |
| **141.** 012122210 | **142.** 101100020 | **143.** 211200020 | **144.** 012010020 | **145.** 110110020 | **146.** 021020020 |
| **147.** 200120020 | **148.** 102220020 | **149.** 110001020 | **150.** 020101020 | **151.** 101011020 | **152.** 122111020 |

TABLE 1—*Continued*

| | | | | | |
|---|---|---|---|---|---|
| **153.** 200211020 | **154.** 121221020 | **155.** 212221020 | **156.** 212102020 | **157.** 120202020 | **158.** 000112020 |
| **159.** 111212020 | **160.** 022212020 | **161.** 011122020 | **162.** 120100120 | **163.** 011100120 | **164.** 202100120 |
| **165.** 210010120 | **166.** 020210120 | **167.** 100020120 | **168.** 012220120 | **169.** 200001120 | **170.** 122201120 |
| **171.** 011011120 | **172.** 002111120 | **173.** 222021120 | **174.** 211121120 | **175.** 020002120 | **176.** 211002120 |
| **177.** 102002120 | **178.** 121012120 | **179.** 220112120 | **180.** 110222120 | | |

**The subcode $\mathcal{D}_{3,9}$.**

| | | | | | |
|---|---|---|---|---|---|
| **181.** 0012221200 | **182.** 2220002200 | **183.** 0102002200 | **184.** 2011102200 | **185.** 1222102200 | **186.** 0020202200 |
| **187.** 1121202200 | **188.** 2212012200 | **189.** 1200112200 | **190.** 1111112200 | **191.** 2021212200 | **192.** 0222212200 |
| **193.** 0211022200 | **194.** 2020122200 | **195.** 1002122200 | **196.** 0100222200 | **197.** 1201222200 | **198.** 0121000010 |
| **199.** 2222000010 | **200.** 1120100010 | **201.** 2201100010 | **202.** 2210200010 | **203.** 1002200010 | **204.** 0112200010 |
| **205.** 1201010010 | **206.** 0102010010 | **207.** 2100110010 | **208.** 0221110010 | **209.** 2012110010 | **210.** 1010210010 |
| **211.** 2001210010 | **212.** 1222210010 | **213.** 0210020010 | **214.** 2020020010 | **215.** 1011020010 | **216.** 0101120010 |
| **217.** 1022120010 | **218.** 2111220010 | **219.** 0202220010 | **220.** 1020001010 | **221.** 0201001010 | **222.** 2111001010 |
| **223.** 2002001010 | **224.** 2010101010 | **225.** 1211101010 | **226.** 0102101010 | **227.** 1100201010 | **228.** 0220201010 |
| **229.** 2021201010 | **230.** 2200011010 | **231.** 0021011010 | **232.** 1212011010 | **233.** 0120111010 | **234.** 1002111010 |
| **235.** 0111211010 | **236.** 2102211010 | **237.** 1110021010 | **238.** 1221021010 | **239.** 0012021010 | **240.** 2122021010 |
| **241.** 2220121010 | **242.** 0000221010 | **243.** 2201221010 | **244.** 2100002010 | **245.** 0011002010 | **246.** 0200102010 |
| **247.** 1001102010 | **248.** 2121102010 | **249.** 2212102010 | **250.** 0022202010 | **251.** 2010012010 | **252.** 1120012010 |
| **253.** 2221012010 | **254.** 1210112010 | **255.** 1101212010 | **256.** 1102022010 | **257.** 0222022010 | **258.** 0110122010 |
| **259.** 2002122010 | **260.** 1200222010 | **261.** 2120222010 | **262.** 1021222010 | **263.** 1200000110 | **264.** 2120000110 |
| **265.** 2011000110 | **266.** 1112000110 | **267.** 0022000110 | **268.** 2000100110 | **269.** 0210100110 | **270.** 1222100110 |
| **271.** 1101200110 | **272.** 0221200110 | **273.** 2202200110 | **274.** 2210010110 | **275.** 1020010110 | **276.** 0002110110 |
| **277.** 0100210110 | **278.** 2112210110 | **279.** 0201020110 | **280.** 1121020110 | **281.** 1100120110 | **282.** 2021120110 |
| **283.** 2212120110 | **284.** 0010220110 | **285.** 1220220110 | **286.** 0122220110 | **287.** 2221001110 | **288.** 0020101110 |
| **289.** 2101101110 | **290.** 1012101110 | **291.** 0011201110 | **292.** 1122201110 | **293.** 2001011110 | **294.** 0200111110 |
| **295.** 2110111110 | **296.** 1221111110 | **297.** 1210211110 | **298.** 2121211110 | **299.** 0222211110 | **300.** 1000021110 |
| **301.** 0120021110 | **302.** 0001121110 | **303.** 1111121110 | **304.** 1202121110 | **305.** 2020221110 | **306.** 0110002110 |
| **307.** 1211002110 | **308.** 1002002110 | **309.** 2220102110 | **310.** 1010202110 | **311.** 2111202110 | **312.** 0212202110 |
| **313.** 2102012110 | **314.** 1000112110 | **315.** 0011112110 | **316.** 2022112110 | **317.** 2200212110 | **318.** 0021022110 |
| **319.** 2201122110 | **320.** 0102122110 | **321.** 1112222110 | **322.** 2222222110 | | |

**The subcode $\mathcal{D}_{3,10}$.**

TABLE 2
The codes $\mathscr{C}_{4,n}$, for $n = 6, 7, 8$.

| | | | | | |
|---|---|---|---|---|---|
| **1.** 111100 | **2.** 222100 | **3.** 333100 | **4.** 321200 | **5.** 132200 | **6.** 213200 |
| **7.** 231300 | **8.** 312300 | **9.** 123300 | **10.** 211010 | **11.** 122010 | **12.** 310110 |
| **13.** 021110 | **14.** 103110 | **15.** 220210 | **16.** 302210 | **17.** 033210 | **18.** 130310 |
| **19.** 131020 | **20.** 313020 | **21.** 120120 | **22.** 201120 | **23.** 012120 | **24.** 330220 |
| **25.** 210320 | **26.** 102320 | **27.** 232030 | **28.** 110230 | **29.** 022230 | **30.** 320330 |
| **31.** 011330 | **32.** 203330 | **33.** 311001 | **34.** 223001 | **35.** 210101 | **36.** 031101 |
| **37.** 102101 | **38.** 120201 | ¿**39.** 201201 | **40.** 012201 | **41.** 330301 | **42.** 110011 |
| **43.** 202011 | **44.** 333011 | **45.** 301111 | **46.** 013111 | **47.** 131211 | **48.** 020311 |
| **49.** 320021 | **50.** 032021 | **51.** 103021 | **52.** 233121 | **53.** 021221 | **54.** 111321 |
| **55.** 222321 | **56.** 121031 | **57.** 130131 | **58.** 312131 | **59.** 300231 | **60.** 002331 |
| **61.** 112002 | **62.** 320102 | **63.** 203102 | **64.** 230202 | **65.** 023202 | **66.** 101302 |
| **67.** 032302 | **68.** 321012 | **69.** 030112 | **70.** 212112 | **71.** 100212 | **72.** 011212 |
| | | **The subcode $\mathscr{D}_{4,6}$.** | | | |
| **73.** 0033120 | **74.** 3020220 | **75.** 3111220 | **76.** 1321220 | **77.** 1132220 | **78.** 3233220 |
| **79.** 2100320 | **80.** 0310320 | **81.** 2211320 | **82.** 2022320 | **83.** 3332320 | **84.** 1223320 |
| **85.** 3220030 | **86.** 1330030 | **87.** 3102030 | **88.** 2203030 | **89.** 3033030 | **90.** 2300130 |
| **91.** 1010130 | **92.** 0120130 | **93.** 3321130 | **94.** 2231130 | **95.** 3113130 | **96.** 2210230 |
| **97.** 3001230 | **98.** 0022230 | **99.** 0303230 | **100.** 0230330 | **101.** 0101330 | **102.** 2112330 |
| **103.** 1032330 | **104.** 1313330 | **105.** 3310001 | **106.** 1130001 | **107.** 1301001 | **108.** 3021001 |
| **109.** 0231001 | **110.** 3200101 | **111.** 0320101 | **112.** 2030101 | **113.** 2311101 | **114.** 3112101 |
| **115.** 1232101 | **116.** 0103101 | **117.** 1013101 | **118.** 2223101 | **119.** 3333101 | **120.** 2300201 |
| **121.** 3211201 | **122.** 2131201 | **123.** 0312201 | **124.** 3032201 | **125.** 0210301 | **126.** 1020301 |
| **127.** 2001301 | **128.** 2122301 | **129.** 1320011 | **130.** 3030011 | **131.** 0302011 | **132.** 3222011 |
| **133.** 1003011 | **134.** 0113011 | **135.** 2333011 | **136.** 3120111 | **137.** 0230111 | **138.** 3301111 |
| **139.** 2322111 | **140.** 3213111 | **141.** 1133111 | **142.** 1311211 | **143.** 0031211 | **144.** 3102211 |
| **145.** 2232211 | **146.** 2013211 | **147.** 3323211 | **148.** 0100311 | **149.** 2310311 | **150.** 0221311 |
| **151.** 0012311 | **152.** 1332311 | **153.** 2203311 | **154.** 3100021 | **155.** 0330021 | **156.** 2321021 |
| **157.** 3012021 | **158.** 0203021 | **159.** 1300121 | | | |
| | | **The subcode $\mathscr{D}_{4,7}$.** | | | |

TABLE 2—*Continued*

| | | | | | |
|---|---|---|---|---|---|
| **160.** 32211210 | **161.** 21321210 | **162.** 20031210 | **163.** 03131210 | **164.** 12302210 | **165.** 33312210 |
| **166.** 11032210 | **167.** 00232210 | **168.** 11103210 | **169.** 22203210 | **170.** 12013210 | **171.** 01313210 |
| **172.** 03223210 | **173.** 33033210 | **174.** 10333210 | **175.** 12000310 | **176.** 20100310 | **177.** 31110310 |
| **178.** 10310310 | **179.** 02120310 | **180.** 11220310 | **181.** 33320310 | **182.** 00011310 | **183.** 12111310 |
| **184.** 30231310 | **185.** 01102310 | **186.** 22012310 | **187.** 03212310 | **188.** 10022310 | **189.** 21232310 |
| **190.** 02332310 | **191.** 30003310 | **192.** 21303310 | **193.** 32313310 | **194.** 23023310 | **195.** 32300020 |
| **196.** 03210020 | **197.** 10320020 | **198.** 11030020 | **199.** 30130020 | **200.** 01301020 | **201.** 00231020 |
| **202.** 33202020 | **203.** 13312020 | **204.** 02322020 | **205.** 12132020 | **206.** 31003020 | **207.** 20103020 |
| **208.** 02013020 | **209.** 33113020 | **210.** 11213020 | **211.** 23323020 | **212.** 23000120 | **213.** 30010120 |
| **214.** 11310120 | **215.** 33120120 | **216.** 22130120 | **217.** 00330120 | **218.** 03101120 | **219.** 13211120 |
| **220.** 20311120 | **221.** 32021120 | **222.** 21031120 | **223.** 30102120 | **224.** 21302120 | **225.** 03012120 |
| **226.** 32212120 | **227.** 01232120 | **228.** 10003120 | **229.** 01113120 | **230.** 12223120 | **231.** 31323120 |
| **232.** 23233120 | **233.** 13100220 | **234.** 31210220 | **235.** 20230220 | **236.** 33330220 | **237.** 30001220 |
| **238.** 23201220 | **239.** 02311220 | **240.** 13321220 | **241.** 12031220 | **242.** 00302220 | **243.** 23112220 |
| **244.** 32122220 | **245.** 01203220 | **246.** 10113220 | **247.** 00023220 | **248.** 21133220 | **249.** 10200320 |
| **250.** 00110320 | **251.** 23310320 | **252.** 20020320 | **253.** 01320320 | **254.** 02230320 | **255.** 02001320 |
| **256.** 11011320 | **257.** 30121320 | **258.** 21221320 | **259.** 33031320 | **260.** 10331320 | **261.** 13002320 |
| **262.** 20212320 | **263.** 31312320 | **264.** 11122320 | **265.** 32103320 | **266.** 03123320 | **267.** 31233320 |
| **268.** 22333320 | **269.** 12100030 | **270.** 21200030 | **271.** 01120030 | **272.** 03330030 | **273.** 30101030 |
| **274.** 13301030 | **275.** 20211030 | **276.** 32311030 | **277.** 11221030 | **278.** 21331030 | **279.** 11002030 |
| **280.** 02202030 | **281.** 20022030 | **282.** 00132030 | **283.** 23232030 | **284.** 22003030 | **285.** 21113030 |
| **286.** 00313030 | **287.** 03023030 | **288.** 10123030 | **289.** 30233030 | **290.** 01300130 | **291.** 02010130 |
| **292.** 33210130 | **293.** 13130130 | **294.** 32330130 | **295.** 22101130 | **296.** 10011130 | **297.** 01021130 |
| **298.** 30321130 | **299.** 12231130 | **300.** 11212130 | **301.** 13022130 | **302.** 21122130 | **303.** 30032130 |
| **304.** 33003130 | **305.** 20303130 | **306.** 32113130 | **307.** 02323130 | **308.** 11033130 | **309.** 10300230 |
| **310.** 11010230 | **311.** 00210230 | **312.** 22120230 | **313.** 31320230 | **314.** 22011230 | **315.** 13111230 |
| **316.** 33021230 | **317.** 02221230 | **318.** 31231230 | **319.** 03002230 | **320.** 20102230 | **321.** 02112230 |
| **322.** 10222230 | **323.** 23322230 | **324.** 33132230 | **325.** 01332230 | **326.** 13203230 | **327.** 32303230 |
| **328.** 30013230 | **329.** 03100330 | **330.** 12210330 | **331.** 30220330 | **332.** 01030330 | **333.** 31001330 |
| **334.** 00301330 | **335.** 01111330 | **336.** 12021330 | | | |

**The subcode $\mathscr{D}_{4,8}$.**